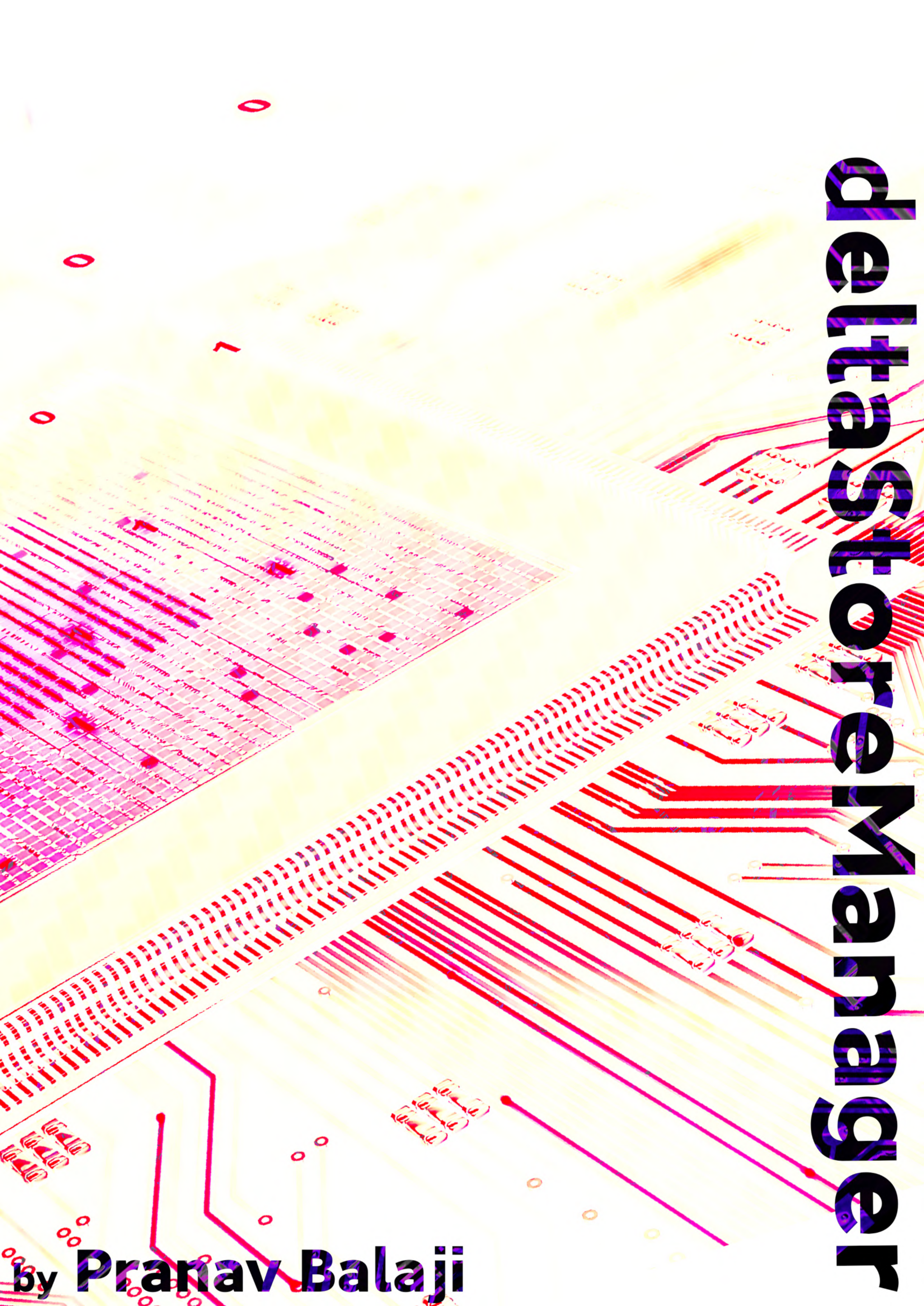


deltaStoreManager

by **Pranav Balaji**

deltaStoreManager

Python Project Report

TOPIC: Store Management

Made by Pranav Balaji

Class: XI – A

ACKNOWLEDGMENT:

This is to certify that Pranav Balaji of Apeejay School, Noida (Class: 11-A) has created a project on the topic “Store Management”, called as the “deltaStoreManager”.

He has generated this report after a lot of hard work. The report has been created under the guidance of the teacher Mrs. Sujata Bhardwaj and qualifies the benchmarks for the Python Project.

ABOUT THE PROJECT:

The project is focussed on being an all-rounder solution for shops to handle their billing, customers and record-keeping needs.

The project asks the user to select from 5 options, namely:

THIS PROJECT INCLUDES:

- BILLING MODE
- NEW CUSTOMER REGISTRATION MODE
- VIEW ALL REGISTERED CUSTOMERS MODE
- VIEW ALL SALES RECORDS MODE
- EXIT PROGRAM OPTION

The same has been discussed in brief-detail below:

In the billing mode, the user is asked to enter the customer's I.D. (if registered), and then asks for the number of items being purchased. Then it asks the user to enter each product's code, followed by the tax and discount amount (can be zero) to be applied. The total amount is subsequently printed.

In the customer registration mode, the user is asked to enter the customer's new I.D., full-name and email contact. The same is inserted into the SQL database.

In the view all registered customers mode, the program contacts the SQL database and displays the complete table of registered customers.

In the view all sales records mode, the program first asks the user to authorize the event by asking for a password (default = "root"). Upon successful authenication, the program opens the sales log externally, with the NotePad application (whichever app the system has installed to open .txt files by default would be used).

Upon selecting the fifth – exit option, the program exits.

FUNCTIONS USED:

from datetime import datetime

- A Python in-built function to display the system's date and time.

import mysql.connector

- A function to connect and interact with SQL databases.

import time

- A Python in-built function used to specify delays in the program.

import os

- A Python in-built function used to open external applications.

def mainmenu()

- Self-defined function to display the main-menu of the menu-driven program.

def inserter()

- Self-defined funciton to take values from the user and insert them into the SQL database selected.

HARDWARE USED:

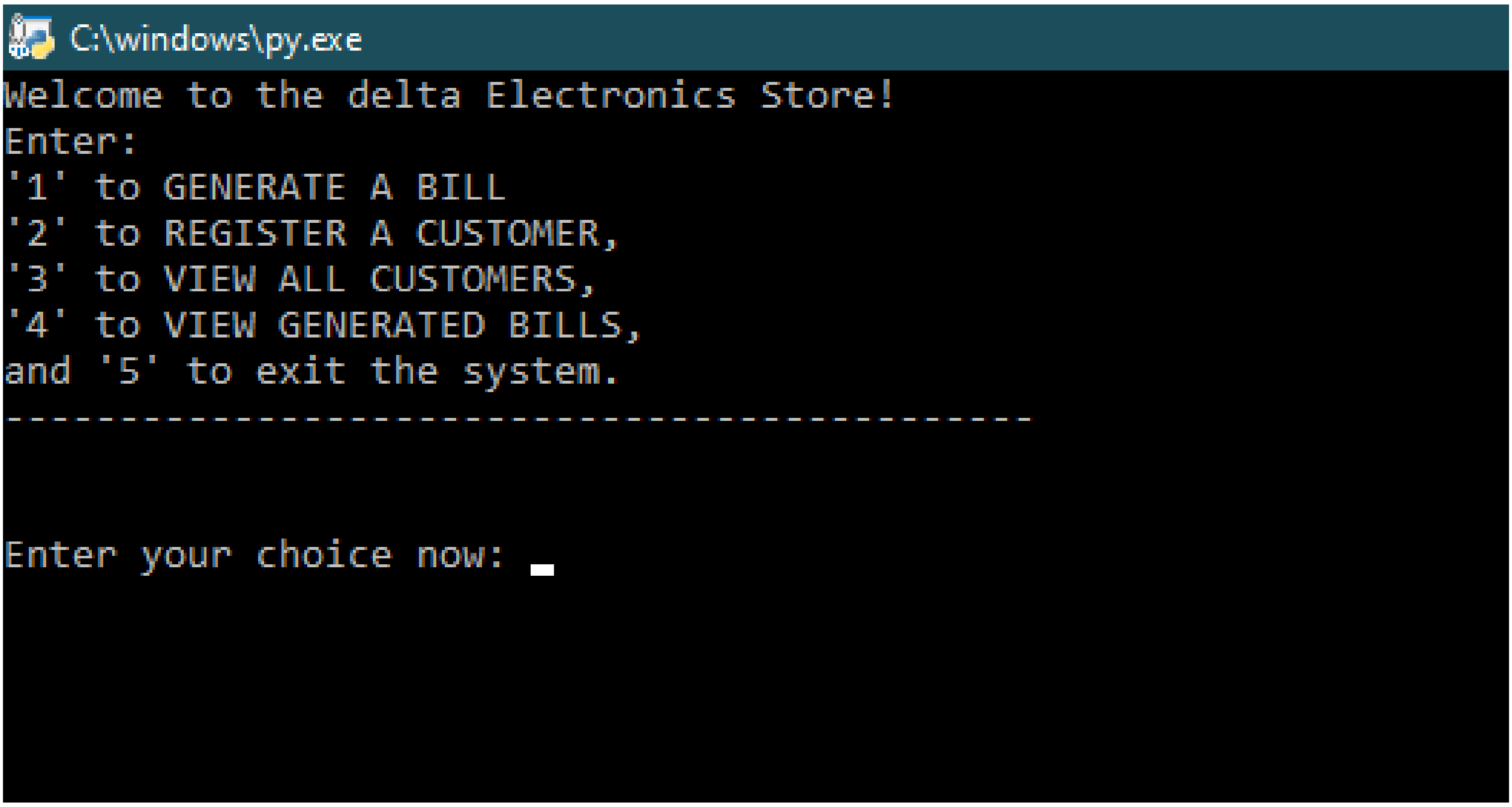
- CPU
- SSD or HDD
- RAM
- NETWORK (if using a non-local database)

SOFTWARE USED:

- Microsoft Windows 10
- Microsoft Office Word
- Microsoft Visual Studio
- GitHub by Microsoft

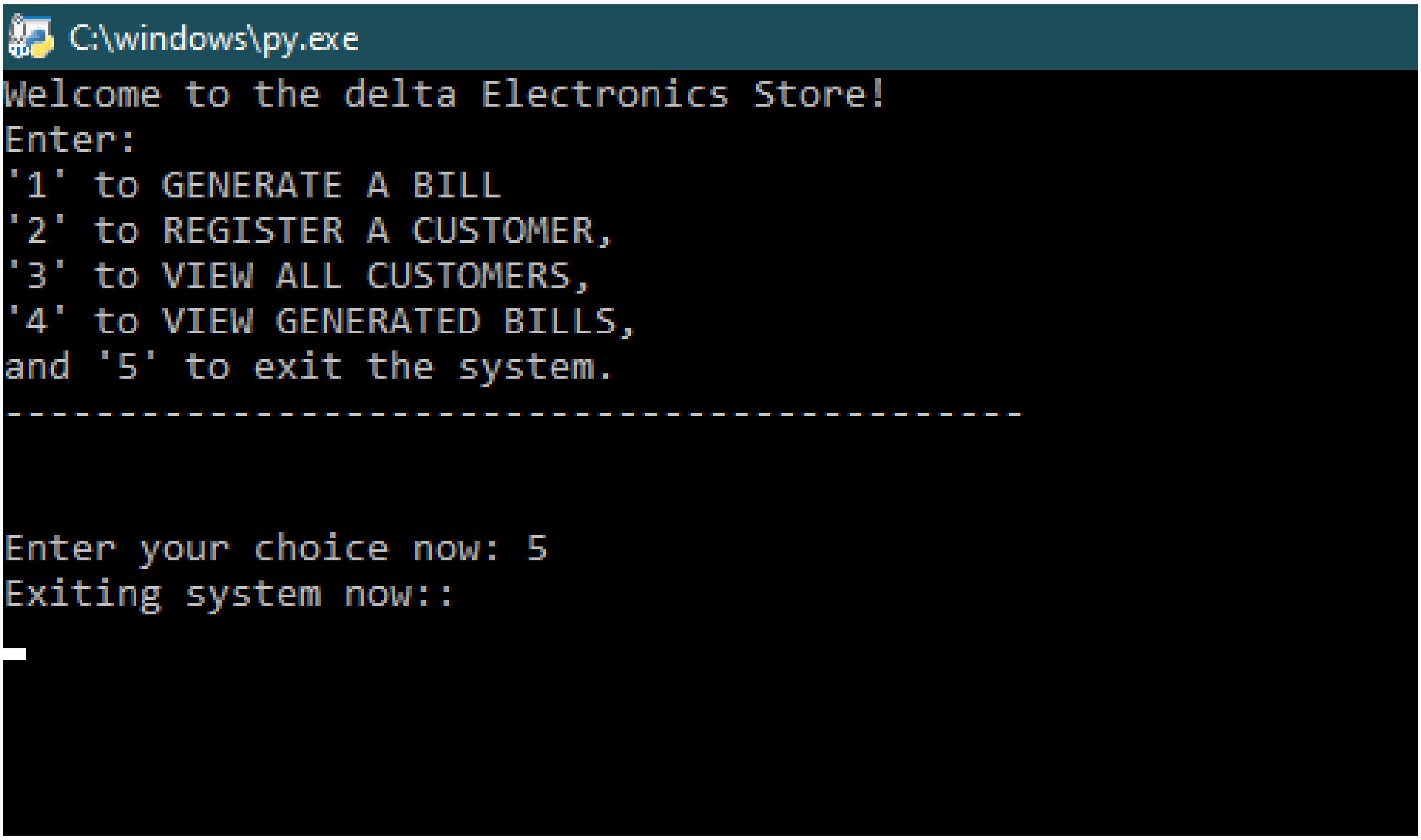
```
                                   ### PROGRAM
# Python-based program for delta Store Manager
# Created by Pranav Balaji; CLASS XI-A
# Created for Class - XI Python Project
# Requires a local SQL database (named delta) with a table (named cust).
# Database can be not-local, i.e. hosted on the internet; values to be speified for the same in the program.

def mainmenu(): #defining a function for the main menu
    print("Welcome to the delta Electronics Store!")
    print("Enter: ")
    print("'1' to GENERATE A BILL")
    print("'2' to REGISTER A CUSTOMER,")
    print("'3' to VIEW ALL CUSTOMERS,")
    print("'4' to VIEW GENERATED BILLS,")
    print("and '5' to exit the system.")
    print("----------------------------------------")
    print()
    print()

from datetime import datetime #for reporting the billing time and date
now = datetime.now()
dt_string = now.strftime("%d/%m/%Y %H:%M:%S") #datetime object containing current date and time
logger = open(r"log.txt","a+") #Opening / creating (if it doesn't exist already) the .txt record file
logger.write("---------------------------------------- \n")
logger.write("deltaStoreManager \n")
logger.write("SALES RECORD: \n")
import mysql.connector #to connect to the SQL database (local)
import time #to provide delays to make the system run seamlessly
import os #library used to open the notepad application to display the sales records
conn = mysql.connector.connect(host='localhost', database='delta', user='root', password='shieldlogmein') #sql connection parameters
cursor = conn.cursor()
cursor.execute("select * from cust")
row = cursor.fetchone()
def inserter(custid, custname, email): #defining a function to input data into the SQL database's table
    conn = mysql.connector.connect(host='localhost', database='delta', user='root', password='shieldlogmein')
    cursor = conn.cursor(buffered=True)
    str = "insert into cust(custid, custname, email) values('%s', '%s', '%s')"
    io = (custid, custname, email)
    cursor.execute(str % io)
    conn.commit()
    print("Customer registered successfully! - deltaDatabaseHandler")

while(1): #while (always) true
    mainmenu() #mainmenu
    time.sleep(1) #for a seamless experience
    decfac = int(input("Enter your choice now: "))

    #Bill Mode
    if decfac == 1:
        print()
        print("Billing MODE: ")
        print()
        custid = input("Enter customer ID if already registered; else press enter: ")
        logger.write("-----------------  ") #writing to log file
        logger.write("Customer ID: \n")
        logger.write(custid)
        logger.write("  \n")
        logger.write("Date and time: \n") #including the date and time of billing (as taken from the system)
        logger.write(dt_string)
        logger.write(" \n")
        abcd1 = 1
        time.sleep(0.7) #for a seamless experience
        #Values stored in two dictionaries
        data = {"del1":40000, "del2":55000, "del3":67000, "del4":25000, "del5":21000, "del6":14000, "del7":13000, "del8":220000, "del9":4500, "del10":17000, "del11":1200, "del12":3700, "del13":4500, "del14":2200, "del15":700, "del16":2750, "del17":6499, "del18":1499, "del19":799, "del20":27000, "del21":6750, "del22":2100, "del23":1199, "del24":3210, "del25":989, "del26":750, "del27":1700, "del28":600, "del29":2175, "del30":890, "del31":2100, "del32":7158, "del33":597, "del34":347, "del35":500, "del36":300, "del37":1097, "del38":80000, "del39":87900, "del40":23790}
        namie = {"del1":"TV 4K OLED 50", "del2":"TV FHD OLED 50", "del3":"8K QLED 80", "del4":"Redmi K20 PRO", "del5":"Redmi K20", "del6":"Redmi Note 8 PRO", "del7":"POCOPHONE F1", "del8":"Mi MIX ALPHA", "del9":"delta CaptureElite Wireless Headphones", "del10":"delta CaptureElite Noise-Cancelling Wireless Headphones", "del11":"delta CaptureElite Essentials Headphones", "del12":"delta CaptureElite Gaming Headphones", "del13":"delta CaptureElite Truly-Wireless Eadphones", "del14":"delta CaptureElite Neckband-Style Wireless Earphones", "del15":"delta CaptureElite Essentials Earphones", "del16":"delta CaptureElite Gaming Earphones", "del17":"delta CaptureElite 30W Bluetooth Speakers", "del18":"delta CaptureElite 10W Bluetooth Speakers", "del19":"delta CaptureElite Essentials Bluetooth Speaker", "del20":"delta CaptureElite ULTRA Home Theatre", "del21":"delta CaptureElite Essentials Home Theatre", "del22":"delta CaptureElite Wired Speaker - 5.1", "del23":"delta CaptureElite Essentials Wired Speaker - STEREO", "del24":"delta Polowski Tactical SHERPAELITE Power Bank 30000mah", "del25":"delta Polowski Tactical Essentials Power Bank 10000mah", "del26":"delta Polowski Tactical Essentials Mouse", "del27":"delta Polowski Tactical RGB Gaming Mouse", "del28":"delta Polowski Tactical Essentials Keyboard", "del29":"delta Polowski Tactical RGB Gaming Keyboard", "del30":"delta Polowski Tactical SHERPAELITE Flashlight", "del31":"deltaNetworking Wi-Fi Router AX17", "del32":"deltaNetworking SHERPAELITE Mesh Wi-Fi Router", "del33":"deltaSupport 120W Laptop Adapter", "del34":"deltaSupport 60W Laptop Adapter", "del35":"deltaSupport Phone Case", "del36":"deltaSupport Essentials Phone Charger 10W", "del37":"deltaSupport SHERPAELITE Phone Charger 30W", "del38":"deltaCiccadella Gaming Laptop", "del39":"deltaCiccadella Content Creator's Laptop", "del40":"deltaCiccadella Student's Laptop"}
        numfac = int(input("Enter the number of items: "))
        time.sleep(1) #for a seamless experience
        afac = 0
        billiemaster = 0 #variable for totalling the price
        while(afac!=numfac):
            item = input("Enter the item code: ")
            time.sleep(1) #for a seamless experience
            if item in data:
                billiemaster+=data[item]
                print("Product purchased: ", namie[item], " costing: ", data[item])
                print("---")
                logger.write("Purchased: \n") #writing to file
                logger.write(namie[item])
                logger.write(" \n")

            else:
                print("Wrong input. Try again!")
                print("---")
            afac+=1
        tax = int(input("Enter the net tax %: "))
        print(tax,"% NET TAX - Incoicing!")
        time.sleep(1) #for a seamless experience
        discount = int(input("Enter the discount %: "))
        print(discount,"% NET DISCOUNT - Invoicing!")
        time.sleep(0.4) #for a seamless experience
        print("Please Wait....... Billing.......")
        time.sleep(1.3) #for a seamless experience
        tota = (((tax/100)*billiemaster)+billiemaster)
        total = tota-((discount/100)*tota)
        print("BILL NUMBER: ", abcd1, "; the total bill is: ", total)
        logger.write("Total amount billed for: \n") #writing to file
        logger.write(str(total))
        logger.write("\n")
        abcd1+=1
        afac+=1
        time.sleep(2) #for a seamless experience
        print()
        print()
    #Register Customer
    elif decfac == 2:
        print("Loading server connection....... ") #SQL connection prompt
        time.sleep(0.4) #for a seamless experience
        custid = input("Enter the customer's customer ID: ")
        custname = input("Enter the customer's name: ")
        email = input("Enter the customer's E-mail ID: ")
        inserter(custid, custname, email) #argumental function to insert values into the SQL database
        print("--------------------------------------")
        time.sleep(1) #for a seamless experience
    #VIEW ALL CUSTOMERS
    elif decfac == 3:
        print()
        print("The registered customers are: ")
        time.sleep(0.7) #for a seamless experience
        #Re-writing to refresh connection
        conn = mysql.connector.connect(host='localhost', database='delta', user='root', password='shieldlogmein')
        cursor = conn.cursor()
        cursor.execute("select * from cust")
        row = cursor.fetchone()
        #takes values from the SQL database
        while row is not None:
            print(row)
            row = cursor.fetchone()
        cursor.close()
        conn.close()
        print()
        print()

    #View Generated Bills
    elif decfac == 4:
        #password verification as sales record is not to be shown to all;
        passw = input("To view all sales records, enter the administrator password: ")
        if passw == "root":
                time.sleep(1) #for a seamless experience
                print("Authorization Succesfull! ")
                print("Opening sales log externally:: ")
                time.sleep(0.6)
                logger.close() #to change file access modes
                logger = open("log.txt","r+")
                # Uncomment the below lines if the program has to be modified to show the records in the shell itself and not externally
                # print(logger.read())
                # print()
                # print("Opening sales log externally now. ")
                time.sleep(1.4) #for a seamless experience
                os.startfile('log.txt') #to open the external notepad application
        else:
            time.sleep(1) #for a seamless experience
            print("Wrong password entered. Try again. ")
            passw = input("To view all sales records, enter the administrator password: ")
            if passw == "root":
                    time.sleep(1) #for a seamless experience
                    print("Authorization Succesfull! ")
                    print("Opening sales log externally:: ")
                    time.sleep(0.6) #for a seamless experience
                    logger.close() #to change file access modes
                    logger = open("log.txt","r+")
                    # print(logger.read())
                    # print()
                    # print("Opening sales log externally now. ")
                    time.sleep(1.4) #for a seamless experience
                    os.startfile('log.txt')
            else:
                print("Multiple Unsuccesfull Attempts Detected. Re-run the program to login now. ")
                time.sleep(1.4) #for a seamless experience
                print()
                print()
    #Exit System
    elif decfac == 5:
        print("Exiting system now:: ")
        time.sleep(0.4) #for a seamless experience
        break
# Program ENDS here
# Available on github: deltaonealpha.github.io/dsmsapl
```

OUTPUT:



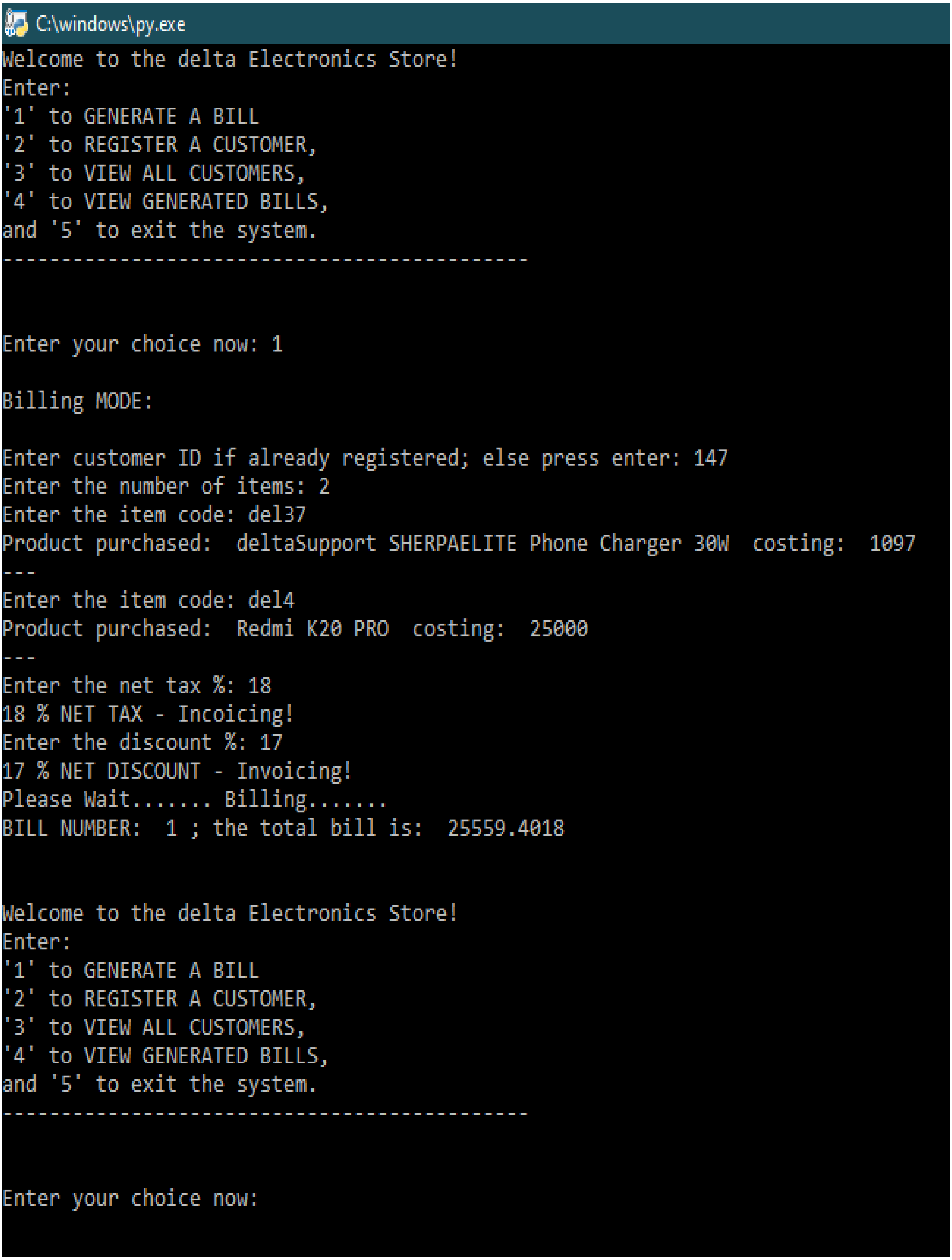
```
Welcome to the delta Electronics Store!
Enter:
'1' to GENERATE A BILL
'2' to REGISTER A CUSTOMER,
'3' to VIEW ALL CUSTOMERS,
'4' to VIEW GENERATED BILLS,
and '5' to exit the system.
---------------------------------------------



Enter your choice now:
```

HOME MENU : Initial boot-time screen



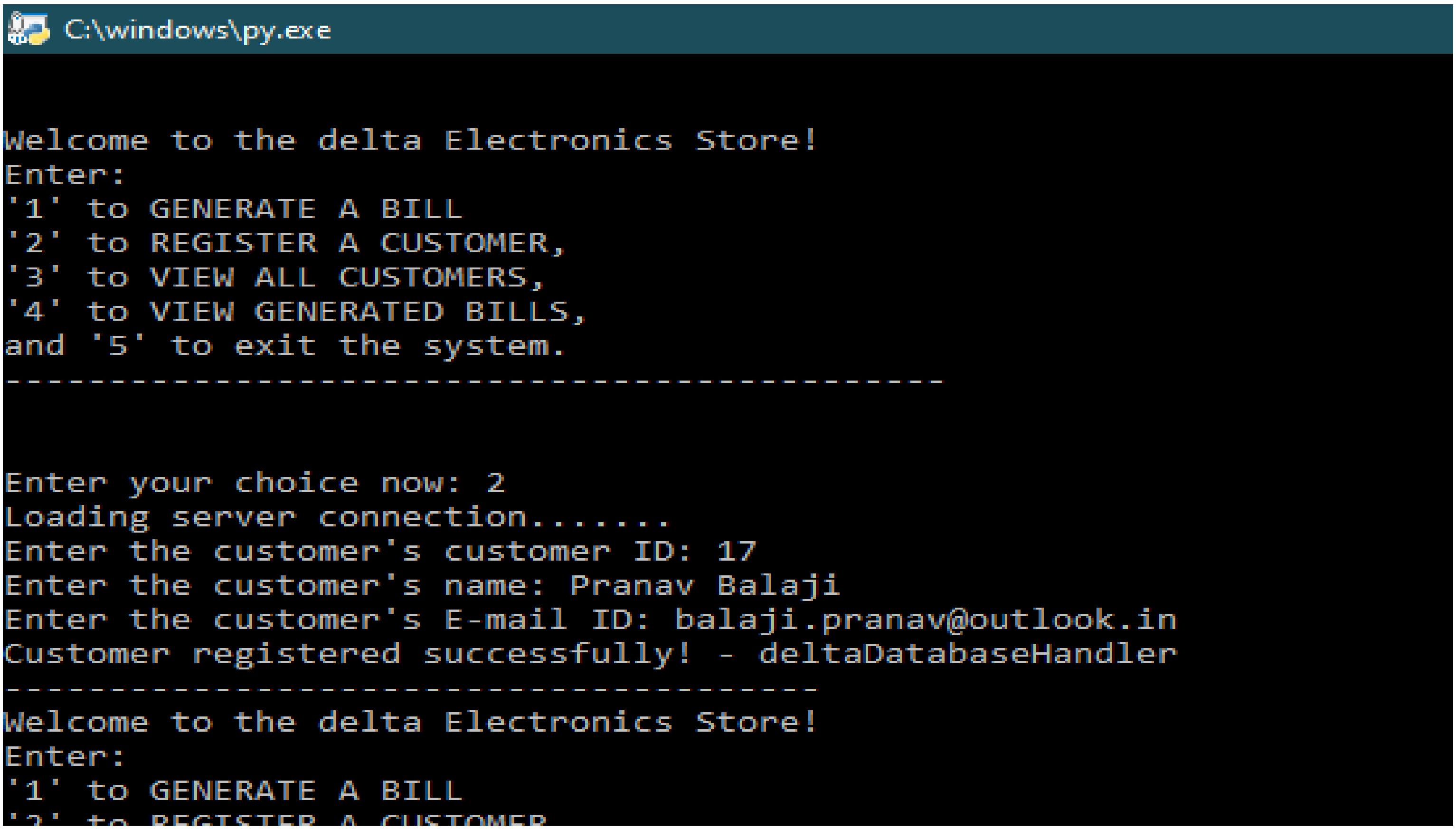
```
Welcome to the delta Electronics Store!
Enter:
'1' to GENERATE A BILL
'2' to REGISTER A CUSTOMER,
'3' to VIEW ALL CUSTOMERS,
'4' to VIEW GENERATED BILLS,
and '5' to exit the system.
---------------------------------------------



Enter your choice now: 5
Exiting system now::
```

EXIT OPTION: Option 5



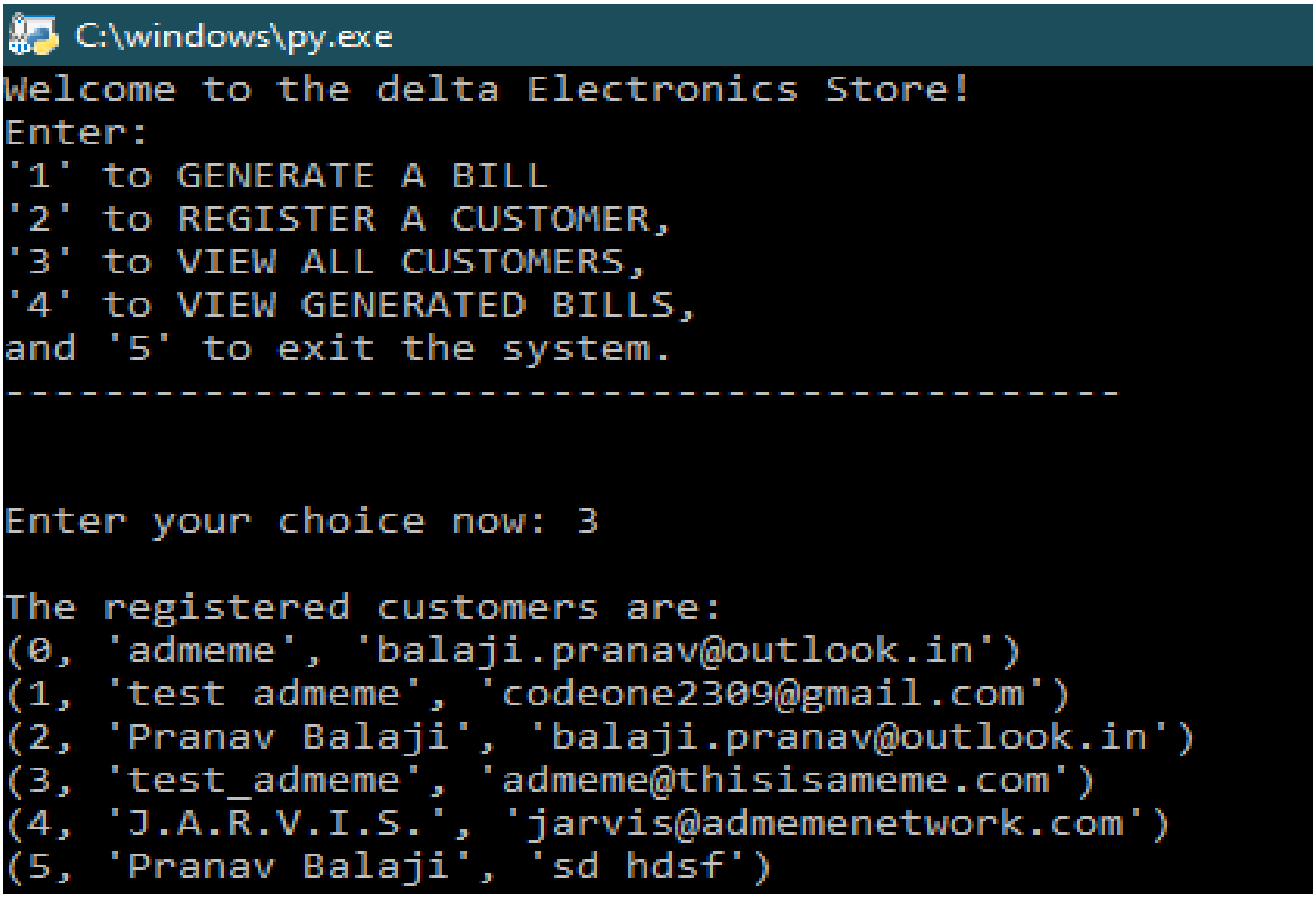
```
Welcome to the delta Electronics Store!
Enter:
'1' to GENERATE A BILL
'2' to REGISTER A CUSTOMER,
'3' to VIEW ALL CUSTOMERS,
'4' to VIEW GENERATED BILLS,
and '5' to exit the system.
---------------------------------------------



Enter your choice now: 1

Billing MODE:

Enter customer ID if already registered; else press enter: 147
Enter the number of items: 2
Enter the item code: del37
Product purchased:  deltaSupport SHERPAELITE Phone Charger 30W  costing:  1097
---
Enter the item code: del4
Product purchased:  Redmi K20 PRO  costing:  25000
---
Enter the net tax %: 18
18 % NET TAX - Incoicing!
Enter the discount %: 17
17 % NET DISCOUNT - Invoicing!
Please Wait....... Billing.......
BILL NUMBER:  1 ; the total bill is:  25559.4018


Welcome to the delta Electronics Store!
Enter:
'1' to GENERATE A BILL
'2' to REGISTER A CUSTOMER,
'3' to VIEW ALL CUSTOMERS,
'4' to VIEW GENERATED BILLS,
and '5' to exit the system.
---------------------------------------------



Enter your choice now:
```

GENERATE A BILL – Option 1



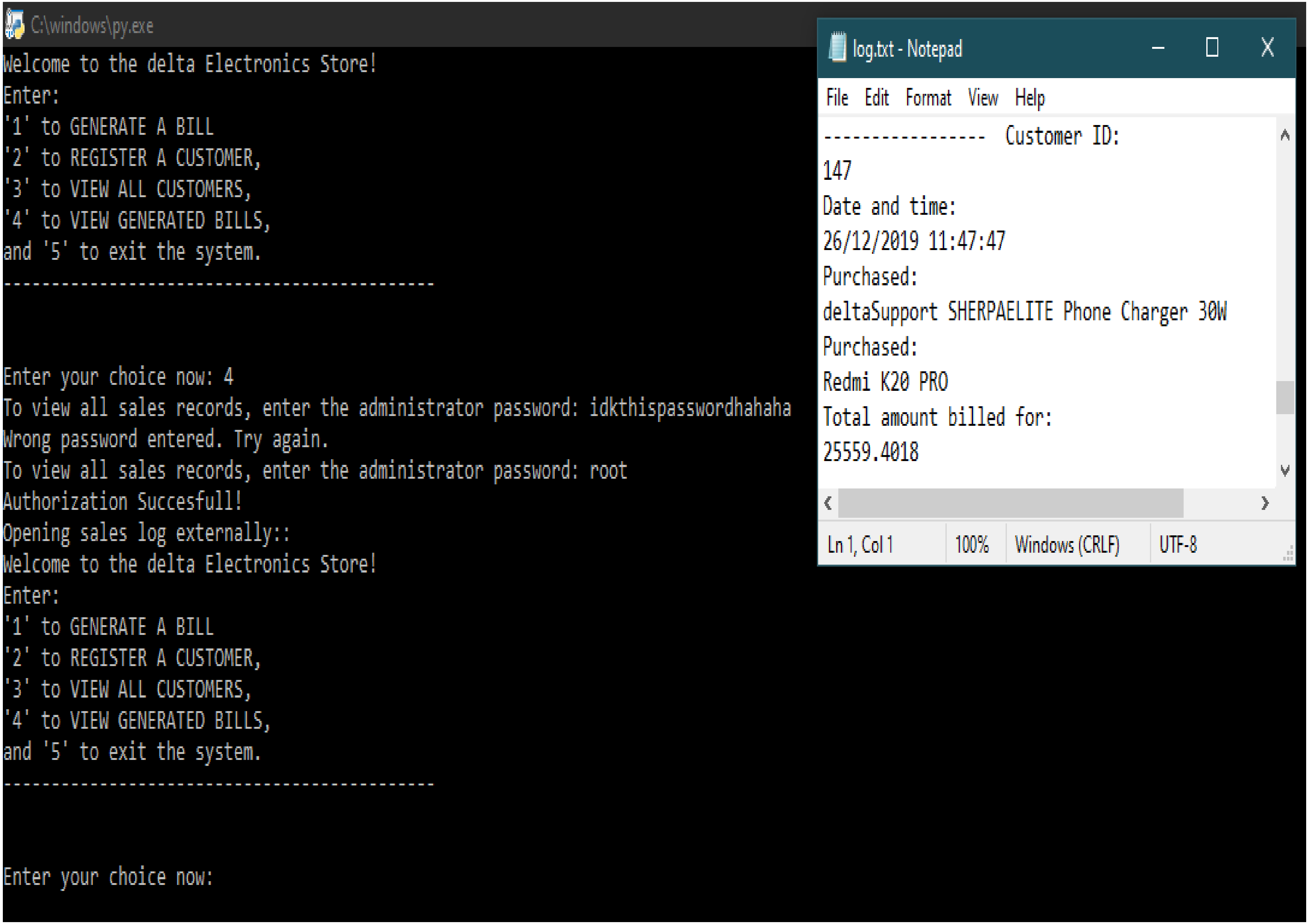
```
Welcome to the delta Electronics Store!
Enter:
'1' to GENERATE A BILL
'2' to REGISTER A CUSTOMER,
'3' to VIEW ALL CUSTOMERS,
'4' to VIEW GENERATED BILLS,
and '5' to exit the system.
---------------------------------------------


Enter your choice now: 2
Loading server connection.......
Enter the customer's customer ID: 17
Enter the customer's name: Pranav Balaji
Enter the customer's E-mail ID: balaji.pranav@outlook.in
Customer registered successfully! - deltaDatabaseHandler
---------------------------------------
Welcome to the delta Electronics Store!
Enter:
'1' to GENERATE A BILL
'2' to REGISTER A CUSTOMER
```

REGISTER A CUSTOMER – Option 2



```
Welcome to the delta Electronics Store!
Enter:
'1' to GENERATE A BILL
'2' to REGISTER A CUSTOMER,
'3' to VIEW ALL CUSTOMERS,
'4' to VIEW GENERATED BILLS,
and '5' to exit the system.
---------------------------------------------


Enter your choice now: 3

The registered customers are:
(0, 'admeme', 'balaji.pranav@outlook.in')
(1, 'test admeme', 'codeone2309@gmail.com')
(2, 'Pranav Balaji', 'balaji.pranav@outlook.in')
(3, 'test_admeme', 'admeme@thisisameme.com')
(4, 'J.A.R.V.I.S.', 'jarvis@admemenetwork.com')
(5, 'Pranav Balaji', 'sd hdsf')
```

VIEW ALL REGISTERED CUSTOMERS – Option 3



```
Welcome to the delta Electronics Store!
Enter:
'1' to GENERATE A BILL
'2' to REGISTER A CUSTOMER,
'3' to VIEW ALL CUSTOMERS,
'4' to VIEW GENERATED BILLS,
and '5' to exit the system.
---------------------------------------------



Enter your choice now: 4
To view all sales records, enter the administrator password: idkthispasswordhahaha
Wrong password entered. Try again.
To view all sales records, enter the administrator password: root
Authorization Succesfull!
Opening sales log externally::
Welcome to the delta Electronics Store!
Enter:
'1' to GENERATE A BILL
'2' to REGISTER A CUSTOMER,
'3' to VIEW ALL CUSTOMERS,
'4' to VIEW GENERATED BILLS,
and '5' to exit the system.
---------------------------------------------



Enter your choice now:
```

```
----------------  Customer ID:
147
Date and time:
26/12/2019 11:47:47
Purchased:
deltaSupport SHERPAELITE Phone Charger 30W
Purchased:
Redmi K20 PRO
Total amount billed for:
25559.4018
```

VIEW ALL GENERATED BILLS (SALES LOG/ RECORD) – Option 4

GitHub:

This program repository has also been uploaded to the code sharing platform, GitHub:

Link to repository: deltaonealpha.github.io/dsmsapl

GitHub DSMSAPL
repository

BIBLIOGRAPHY:

Works sighted from:

- StackOverflow: www.stackoveflow.com
- Google: www.google.com
- Core Python Programming by R. Nageshwar Rao

Thank You

Created by Pranav Balaji

Python Project Class – XI