# PYTHON MINIPROJECT

TOPIC: E COMMERCE SYSTEM

CLASS : SE B1

TEAM MEMBERS:

| NAME | ROLL NO. |
|---|---|
| ANAGHA PATIL | 408 |
| SHIVPRASAD POOJARY | 418 |

## ABSTRACT:

To develop an E-Commerce System wherein, a user would be able to select the products by initially completing the login process and further place the order by entering the payment credentials.

## TECHNOLOGY USED:

The E-Commerce System was made using the following tools:

1] Front-End: Python

2] Back-End- Microsoft Excel

# USER DEFINED FUNCTIONS:

The E-Commerce Website starts with the registration details like: First Name, Last Name, D.O.B., Mobile Number, Gender, E-Mail ID, Username and Passcode. If the username is already taken, the it raises an error : "THE ENTERED USER NAME ALREADY EXISTS IN OUR DATABASE...PLEASE ENTER THE USER NAME AGAIN..." and if the password is less than 8 characters, it raises an error: "INVALID INPUT...PASSCODE SHOULD CONTAIN MINIMUM 8 CHARACTERS". The data is stored in the Excel sheet which is used as database.

## 1] login():

The login page displays two choices : 1. LOGIN   2. SIGNUP

If choice is 1; then the user will be asked for the username and passcode. If either of them is wrong, then it will raise an exception : "INVALID USERNAME AND PASSCODE...PLEASE TRY AGAIN...", otherwise will move to the next page.

If the choice is 2; then the user will move to the signup page.

If the choice is wrong, then an exception will be raised: "INVALID INPUT...PLEASE TRY AGAIN"

## 2] twilio():

The Twilio Python Helper Library makes it easy to interact with the Twilio API from the Python application. Using Twilio's REST API, we can send outgoing SMS messages from our Twilio phone number to mobile phones around the globe. In this function, an one time password is sent to the user's registered mobile number from the admin's mobile number.

## 3] feedback():

The user is asked to give the transaction id sent to their registered email. If the id matches with the one in the database then the list of products will be displayed which was bought by user, else if the id does not match, an error is displayed: "INVALID...ENTER AGAIN...OR ENTER 0 TO EXIT"

The user is asked to enter the product number and the rating out of 5. If the rating is beyond the given range then the error displayed is: "WRONG INPUT !!!..PLEASE ENTER A NUMERIC VALUE OUT OF 5 ". The user is asked to enter the review title and feedback. All the details are displayed once given by the user and the number of reviews and average rating are updated in the database for the particular product.

## 3] check_valid_c():

This function checks whether the product selected by the user was earlier cancelled or not. To check if it is cancelled, we match it with previous transaction id's to get the result. If a cancelled product is selected then an error message is displayed: "INVALID ACTION...THE TRANSACTION HAS ALREADY BEEN CANCELLED REQUEST IS DENIIED"
If the user had cancelled a particular product and if the user selects the cancelled product, then an error is displayed: "INVALID ACTION...THE PRODUCT HAS ALREADY BEEN CANCELLED REQUEST IS DENIED".

## 4] smtp1():

This function displays the confirmation for cancellation of the product by the user and the refund of the product cancelled.

## 5] req_cancel():

The user is asked to provide the transaction id sent to their registered e-mail. If the transaction id is correct it will proceed with the cancellation or else it will print an error: “INVALID...ENTER AGAIN...OR ENTER 0 TO EXIT”. The products bought by the user will be displayed and the confirmation message will be printed : ENTER 1 TO CONFIRM CANCELLATION OF THE ORDER”. The user is then asked to give the product number to be cancelled from the order or else ‘0’ to cancel the entire order. Hence, the order is cancelled, according to the user’s input.

## 6] email():

In this function we import smtplib, which is a module that defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP listener daemon. The central class in the email package is the EmailMessage class, imported from the email.message module. It is the base class for the email object model. EmailMessage provides the core functionality for setting and querying header fields, for accessing message bodies, and for creating or modifying structured messages. In this function, one time password is sent on the user’s registered e-mail id for completing the payment process.

## 7] otp():

In this function, one time password is generated. It is done by importing numpy. We use the built-in function ‘np.random.randint’ which generates a random number of 6 digits and is sent to the user via e-mail.

## 8] bill():

This function is used to create an invoice of the products that the user has purchased. We import pandas and from datetime we import date and datetime. The customer name, customer id, current date and time and transaction id is displayed. The products to be purchased by the user are displayed along with their product id, product price, quantity, total price and savings. If the user purchases more quantity than available stock, an error message is displayed: “WE ARE SORRY… CURRENTLY THE PRODUCT IS OUT OF STOCK...”. The user is given the choice to either purchase the minimum quantity available or else cancel the order. The final amount and the total savings are displayed. The user proceeds for payment after entering ‘1’. If the debit card number is saved then the user is asked to enter the CVV number. If the CVV does not match then the error is displayed: “INVALID...ENTER THE CVV AGAIN…”. If correct, then the one time password is sent to the registered mobile number. If the user is shopping for the first time, then the user needs to enter the debit card details along with the CVV and the details are saved in the database for future shopping (if user permits). The user has to enters the one time password sent on the registered mobile number within 100 seconds. If the one time password entered is incorrect then the error is displayed: “ENTER THE OTP AGAIN:”. The user is given 2 more chances to enter the correct one time password, if the chances exceed, then the transaction fails. If the one time password matches, then we proceed with the transaction and it is added in the database. Hence, a confirmation message is displayed: “ORDER IS PLACED !!!..ORDER CONFIRMATION WITH TRANSACTION ID IS SENT
TO THE REGISTERED G-MAIL ID. THANKYOU FOR SHOPPING WITH US".

## 9] display_sorted():

This function displays all the products in the sorted order along with the actual price, rating and the savings on each product.

## 10] filter_products():

After all the products are displayed, the user is given the choice to apply the filters. If the user selects ‘1’: then the products are displayed according to their prices ranging from low to high.
If the user selects ‘2’: then the products are displayed according to their prices ranging from high to low.

If the user selects ‘3’: then the products are displayed according to their ratings (high to low) If the user selects the wrong choice, then the error is displayed: “INVALID INPUT !!!...”

## 11] fetch():

After the display of all the products, the user is asked to apply the filters or not. If the user enters ‘0’, then no filter is applied. If the user enters ‘1’, then the options of various filters are available. According to the choice entered by the user, the filter is applied. If the user wants to see a specific product, then details of the product are displayed upon entering the product number. The user can see the reviews by entering ‘1’ and all the reviews are displayed along with their respective rating. After displaying all the reviews, the user is asked to enter the choice:
1. BUY NOW
2. ADD TO CART
3. GO BACK

If the choice is ‘3’: then it goes back to the previous page. If the choice is ‘1’: then the user is asked to enter the quantity and it directly goes to the billing page.
If the choice is ‘2’: then the user is asked to enter the quantity and to whether continue shopping or to checkout.

## 12] check_my_rec():

In this function, we have three conditions. In the first case, if the user has viewed the product in the past, then we will recommend the products, which are viewed maximum number of times, related to it. In the second case, If the user hasn't viewed any product then if he has any transaction history, then we will display the products related to it. In the third case, if the user doesn't satisfy any of the conditions, then recommendation will display the products which are viewed by other customers for maximum number of times.

## 13] shop_now():

The function displays various options for the user to select for shopping. Such as:

1. Mobiles, Computers
2. TV, Appliances, Electronics
3. Men's Fashion
4. Women's Fashion
5. Home, Kitchen, Pets, Furniture
6. Grocery and Household appliance
7. Sports, Fitness, Bags, Luggage
8. Toys, Kid's Fashion
9. Car, Motorbike, Industrial
10. Books and Audible

If the user selects ‘1’: The sub categories displayed are:

1. Mobiles
2. Mobile Accessories
3. Power Banks
4. Laptops
5. Tablets
6. Computers and Accessories
7. Office Supplies & Stationary

If the user selects ‘2’: The sub categories displayed are:

1. Televisions
2. Washing machines
3. Refrigerators
4. Microwaves
5. Air Conditioners
6. Dishwashers
7. Home Entertainment systems

If the user selects ‘3’: The sub categories displayed are:

1.Clothing
2.Shoes
3.Watches
4.Bags and Wallets
5.Sunglasses
6.Sportswear

If the user selects ‘4’: The sub categories displayed are:

1.Clothing
2.Shoes
3.Watches
4.Fashion and Silver Jewellery
5.Gold and Diamond Jewellery
6.Handbags and Clutches
7.Sunglasses
8.Sportswear

If the user selects ‘5’: The sub categories displayed are:

1.Kitchen and Dining
2.Home Furnishing
3.Home Décor
4.Garden and Outdoors
5.Indoor Lighting
6.Pet Supplies
7.Art and Craft Supplies

If the user selects ‘6’: The sub categories displayed are:

1.Super Saver Pantry
2.Cooking Essentials
3.Snacks and Beverages
4.Packaged Foods
5.Gourmet Foods

If the user selects ‘7’: The sub categories displayed are:

1.Exercise and Fitness
2.Cycling
3.Badminton
4.Cricket
5.Football
6.Other games
7.Backpacks
8.Travel Luggage

If the user selects ‘8’: The sub categories displayed are:

1.All Toys and Games

2.Kid's Clothing
3.Kid's Shoes
4.School Bags
If the user selects ‘9’: The sub categories displayed are:

1.Motorbike Accessories and Parts
2.Car Accessories
3.Car and Motorbike Care
4.Industrial and Scientific Supplies
5.Lab and Scientific

If the user selects ‘10’: The sub categories displayed are:

1.Fiction Books
2.Comic books
3.School Textbooks
4.Exam Central
5.Textbooks
6.Indian Language Books

If the user enters the wrong input then the error is displayed: “INVALID INPUT !!!...”. After selecting the sub-category, the products of that category are displayed. This is done by calculating the product_id of that category and fetching it from the database.

## 14] head_pg():

The program starts with the head_pg() function wherein the user is given the choice of selecting the following options:

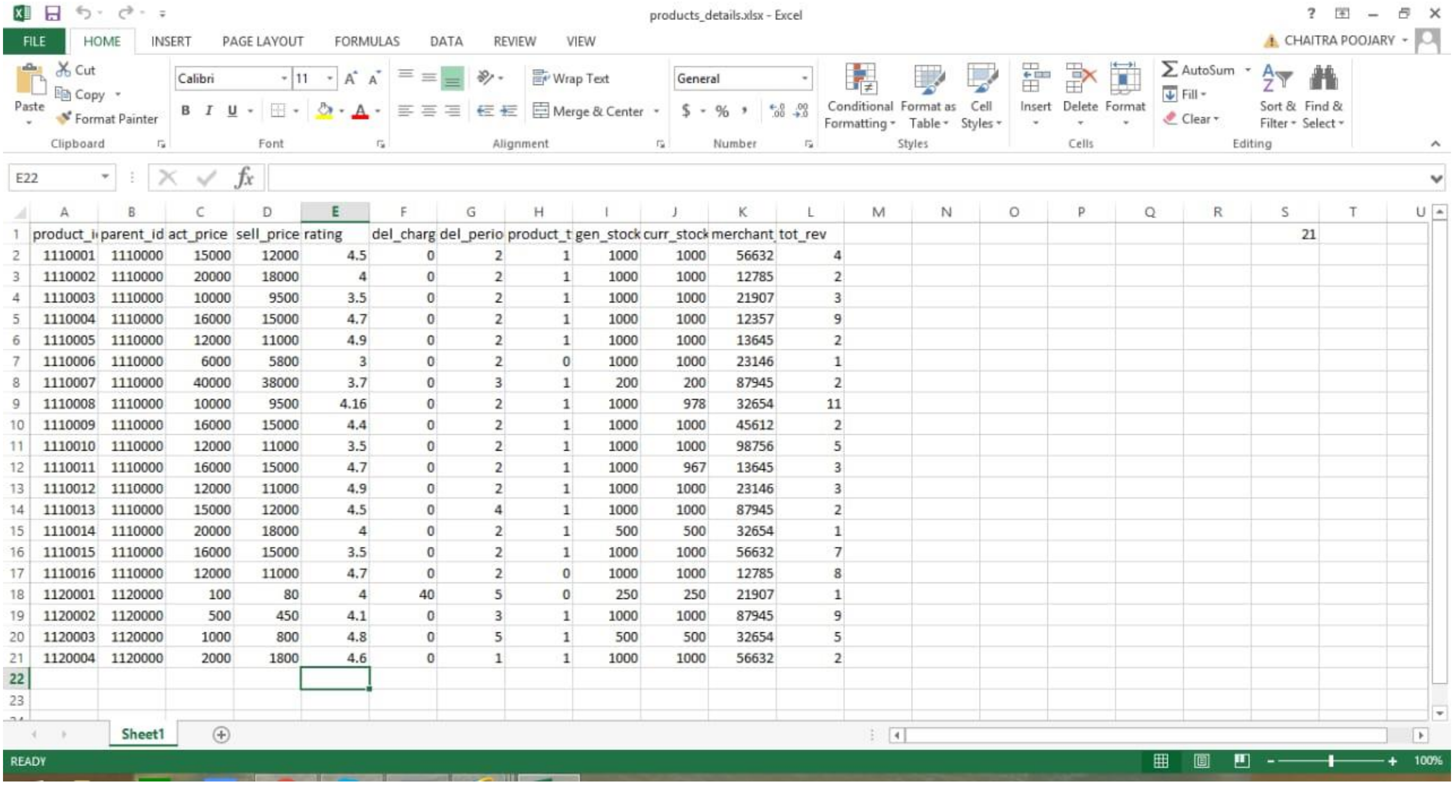
1. CHECK MY RECOMMENDATIONS
2. SHOP NOW
3. REQUEST CANCELLATION
4. POST A REVIEW

The option chosen will take the user to the respective page. If chosen wrong, the message displayed will be: “INVALID INPUT...PLEASE TRY AGAIN..."

# Database:

## 1] Product Details:

This Database contains all the details of the product. Such as: product id, parent id, actual price, delivery charge, delivery period, general stock, current stock, merchant id and total number of reviews got for each product.

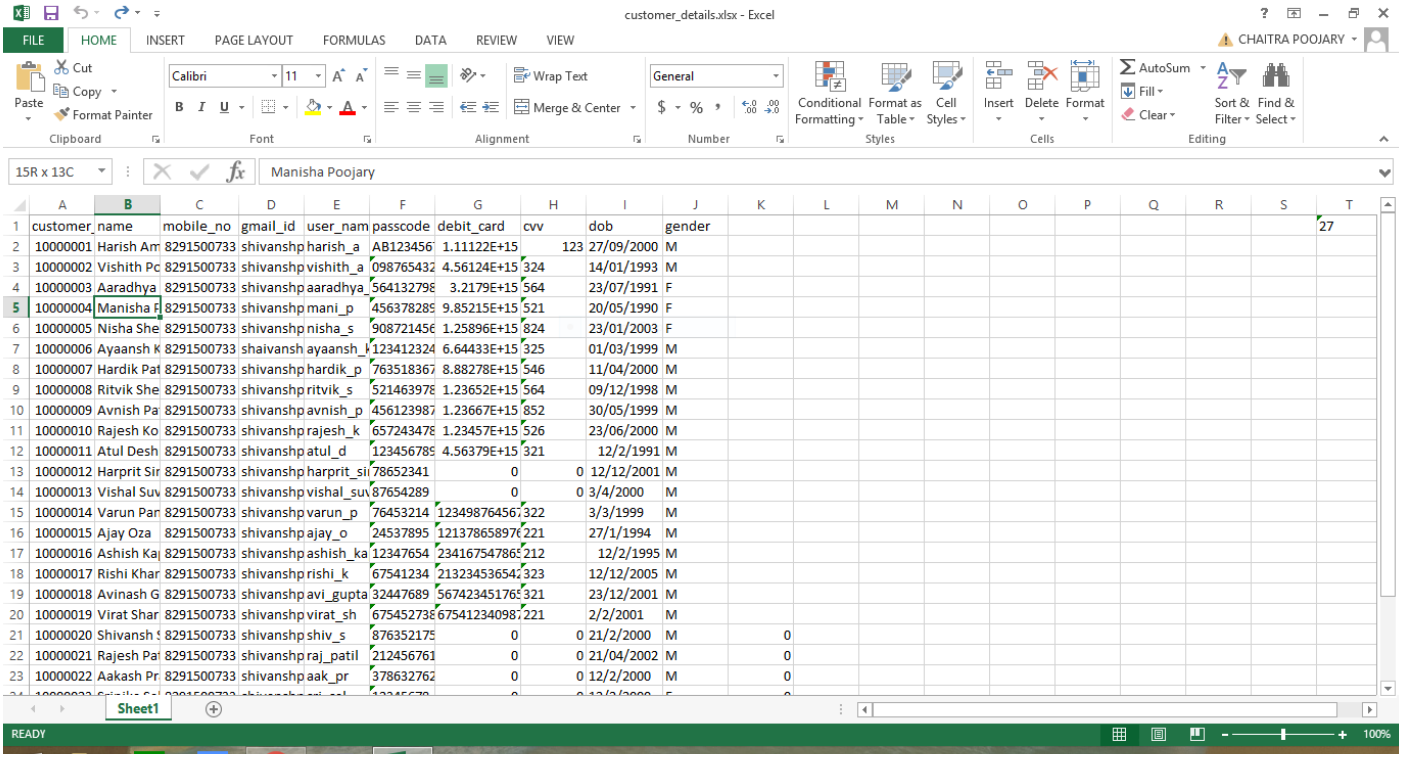


| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | product_i | parent_id | act_price | sell_price | rating | del_charg | del_perio | product_t | gen_stock | curr_stock | merchant | tot_rev |
| 2 | 1110001 | 1110000 | 15000 | 12000 | 4.5 | 0 | 2 | 1 | 1000 | 1000 | 56632 | 4 |
| 3 | 1110002 | 1110000 | 20000 | 18000 | 4 | 0 | 2 | 1 | 1000 | 1000 | 12785 | 2 |
| 4 | 1110003 | 1110000 | 10000 | 9500 | 3.5 | 0 | 2 | 1 | 1000 | 1000 | 21907 | 3 |
| 5 | 1110004 | 1110000 | 16000 | 15000 | 4.7 | 0 | 2 | 1 | 1000 | 1000 | 12357 | 9 |
| 6 | 1110005 | 1110000 | 12000 | 11000 | 4.9 | 0 | 2 | 1 | 1000 | 1000 | 13645 | 2 |
| 7 | 1110006 | 1110000 | 6000 | 5800 | 3 | 0 | 2 | 0 | 1000 | 1000 | 23146 | 1 |
| 8 | 1110007 | 1110000 | 40000 | 38000 | 3.7 | 0 | 3 | 1 | 200 | 200 | 87945 | 2 |
| 9 | 1110008 | 1110000 | 10000 | 9500 | 4.16 | 0 | 2 | 1 | 1000 | 978 | 32654 | 11 |
| 10 | 1110009 | 1110000 | 16000 | 15000 | 4.4 | 0 | 2 | 1 | 1000 | 1000 | 45612 | 2 |
| 11 | 1110010 | 1110000 | 12000 | 11000 | 3.5 | 0 | 2 | 1 | 1000 | 1000 | 98756 | 5 |
| 12 | 1110011 | 1110000 | 16000 | 15000 | 4.7 | 0 | 2 | 1 | 1000 | 967 | 13645 | 3 |
| 13 | 1110012 | 1110000 | 12000 | 11000 | 4.9 | 0 | 2 | 1 | 1000 | 1000 | 23146 | 3 |
| 14 | 1110013 | 1110000 | 15000 | 12000 | 4.5 | 0 | 4 | 1 | 1000 | 1000 | 87945 | 2 |
| 15 | 1110014 | 1110000 | 20000 | 18000 | 4 | 0 | 2 | 1 | 500 | 500 | 32654 | 1 |
| 16 | 1110015 | 1110000 | 16000 | 15000 | 3.5 | 0 | 2 | 1 | 1000 | 1000 | 56632 | 7 |
| 17 | 1110016 | 1110000 | 12000 | 11000 | 4.7 | 0 | 2 | 0 | 1000 | 1000 | 12785 | 8 |
| 18 | 1120001 | 1120000 | 100 | 80 | 4 | 40 | 5 | 0 | 250 | 250 | 21907 | 1 |
| 19 | 1120002 | 1120000 | 500 | 450 | 4.1 | 0 | 3 | 1 | 1000 | 1000 | 87945 | 9 |
| 20 | 1120003 | 1120000 | 1000 | 800 | 4.8 | 0 | 5 | 1 | 500 | 500 | 32654 | 5 |
| 21 | 1120004 | 1120000 | 2000 | 1800 | 4.6 | 0 | 1 | 1 | 1000 | 1000 | 56632 | 2 |

21

## 2] Customer details:

This excel sheet is used to store all the details like customer id, customer name, date of birth, gender, mobile number, g-mail id, user name, passcode, debit card number and cvv.



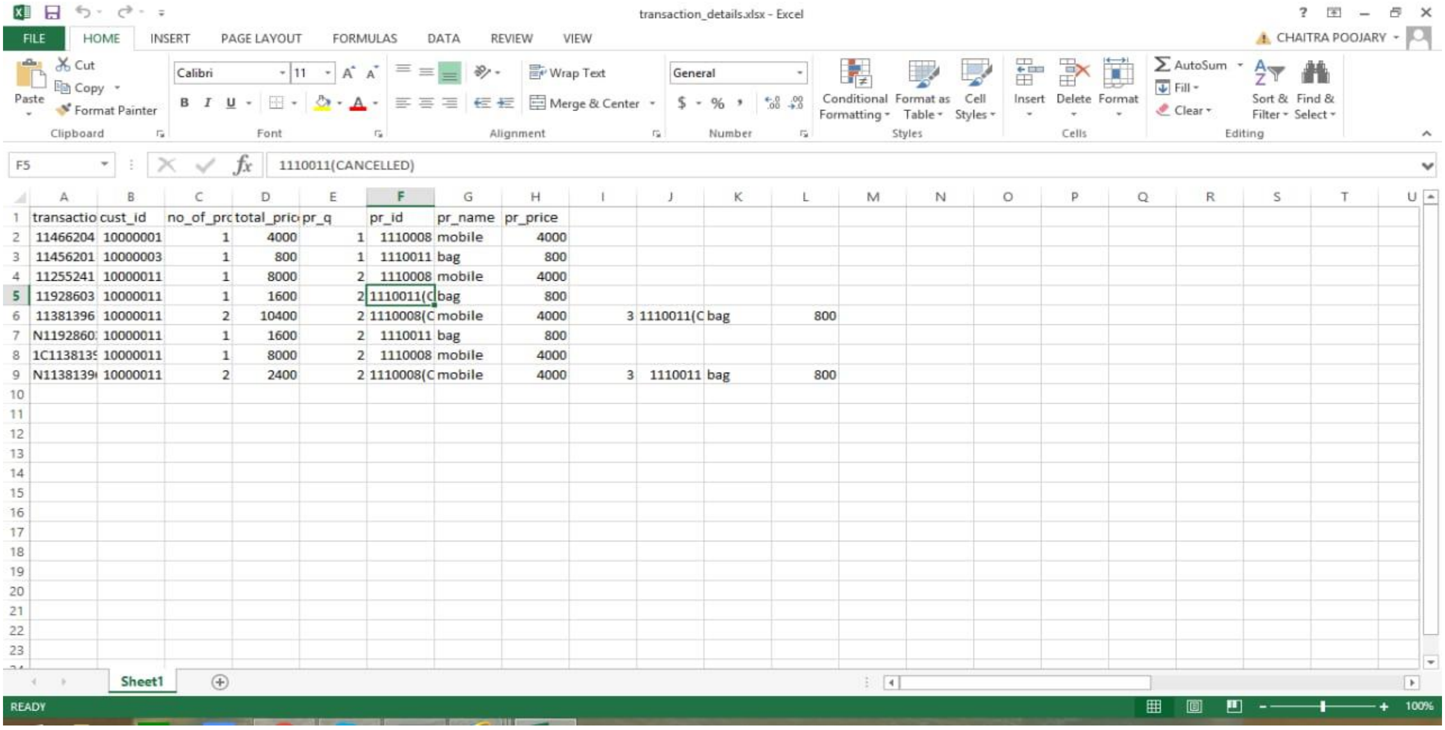
Manisha Poojary

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | customer_ | name | mobile_no | gmail_id | user_nam | passcode | debit_card | cvv | dob | gender | |
| 2 | 10000001 | Harish Am | 8291500733 | shivanshp | harish_a | AB123456 | 1.11122E+15 | 123 | 27/09/2000 | M | |
| 3 | 10000002 | Vishith Pc | 8291500733 | shivanshp | vishith_a | 098765432 | 4.56124E+15 | 324 | 14/01/1993 | M | |
| 4 | 10000003 | Aaradhya | 8291500733 | shivanshp | aaradhya_ | 564132798 | 3.2179E+15 | 564 | 23/07/1991 | F | |
| 5 | 10000004 | Manisha F | 8291500733 | shivanshp | mani_p | 456378289 | 9.85215E+15 | 521 | 20/05/1990 | F | |
| 6 | 10000005 | Nisha She | 8291500733 | shivanshp | nisha_s | 908721456 | 1.25896E+15 | 824 | 23/01/2003 | F | |
| 7 | 10000006 | Ayaansh K | 8291500733 | shaivansh | ayaansh_k | 123412324 | 6.64433E+15 | 325 | 01/03/1999 | M | |
| 8 | 10000007 | Hardik Pat | 8291500733 | shivanshp | hardik_p | 763518367 | 8.88278E+15 | 546 | 11/04/2000 | M | |
| 9 | 10000008 | Ritvik She | 8291500733 | shivanshp | ritvik_s | 521463978 | 1.23652E+15 | 564 | 09/12/1998 | M | |
| 10 | 10000009 | Avnish Pa | 8291500733 | shivanshp | avnish_p | 456123987 | 1.23667E+15 | 852 | 30/05/1999 | M | |
| 11 | 10000010 | Rajesh Ko | 8291500733 | shivanshp | rajesh_k | 657243478 | 1.23457E+15 | 526 | 23/06/2000 | M | |
| 12 | 10000011 | Atul Desh | 8291500733 | shivanshp | atul_d | 123456789 | 4.56379E+15 | 321 | 12/2/1991 | M | |
| 13 | 10000012 | Harprit Sir | 8291500733 | shivanshp | harprit_si | 78652341 | 0 | 0 | 12/12/2001 | M | |
| 14 | 10000013 | Vishal Suv | 8291500733 | shivanshp | vishal_suv | 87654289 | 0 | 0 | 3/4/2000 | M | |
| 15 | 10000014 | Varun Par | 8291500733 | shivanshp | varun_p | 76453214 | 123498764567 | 322 | 3/3/1999 | M | |
| 16 | 10000015 | Ajay Oza | 8291500733 | shivanshp | ajay_o | 24537895 | 121378658976 | 221 | 27/1/1994 | M | |
| 17 | 10000016 | Ashish Ka | 8291500733 | shivanshp | ashish_ka | 12347654 | 234167547865 | 212 | 12/2/1995 | M | |
| 18 | 10000017 | Rishi Khar | 8291500733 | shivanshp | rishi_k | 67541234 | 213234536542 | 323 | 12/12/2005 | M | |
| 19 | 10000018 | Avinash G | 8291500733 | shivanshp | avi_gupta | 32447689 | 567423451765 | 321 | 23/12/2001 | M | |
| 20 | 10000019 | Virat Shar | 8291500733 | shivanshp | virat_sh | 675452738 | 675412340987 | 221 | 2/2/2001 | M | |
| 21 | 10000020 | Shivansh S | 8291500733 | shivanshp | shiv_s | 876352175 | 0 | 0 | 21/2/2000 | M | 0 |
| 22 | 10000021 | Rajesh Pat | 8291500733 | shivanshp | raj_patil | 212456761 | 0 | 0 | 21/04/2002 | M | 0 |
| 23 | 10000022 | Aakash Pr | 8291500733 | shivanshp | aak_pr | 378632762 | 0 | 0 | 12/2/2000 | M | 0 |

27

## 3] Transaction Details:

This Database contains all the details of the transaction history. Such as: transaction id, customer id, number of products purchased, total price, quantity per product, product id, product name and product price.



transaction_details.xlsx - Excel

F5 1110011(CANCELLED)

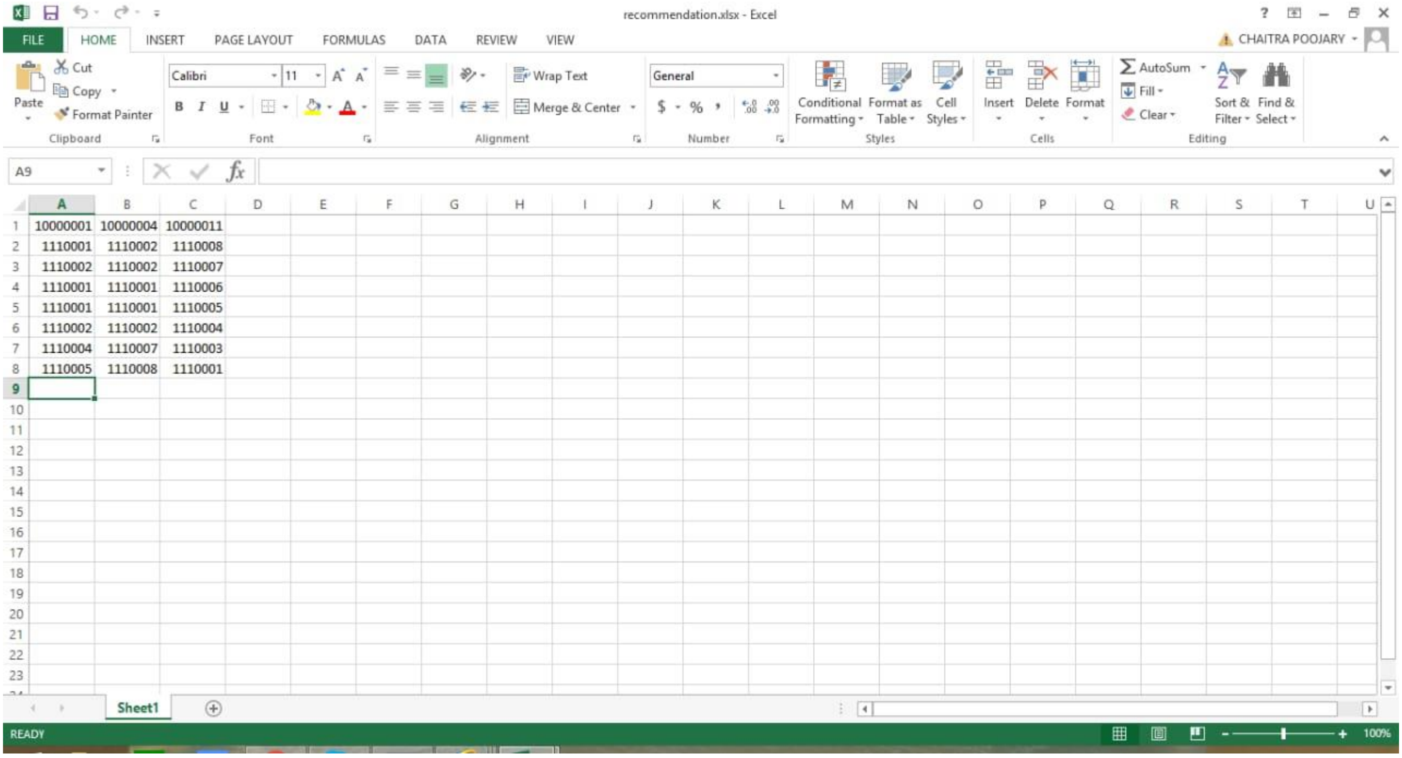
|  | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | transactio | cust_id | no_of_prc | total_pric | pr_q | pr_id | pr_name | pr_price |  |  |  |  |
| 2 | 11466204 | 10000001 | 1 | 4000 | 1 | 1110008 | mobile | 4000 |  |  |  |  |
| 3 | 11456201 | 10000003 | 1 | 800 | 1 | 1110011 | bag | 800 |  |  |  |  |
| 4 | 11255241 | 10000011 | 1 | 8000 | 2 | 1110008 | mobile | 4000 |  |  |  |  |
| 5 | 11928603 | 10000011 | 1 | 1600 | 2 | 1110011(C | bag | 800 |  |  |  |  |
| 6 | 11381396 | 10000011 | 2 | 10400 | 2 | 1110008(C | mobile | 4000 | 3 | 1110011(C | bag | 800 |
| 7 | N1192860 | 10000011 | 1 | 1600 | 2 | 1110011 | bag | 800 |  |  |  |  |
| 8 | 1C113813 | 10000011 | 1 | 8000 | 2 | 1110008 | mobile | 4000 |  |  |  |  |
| 9 | N1138139 | 10000011 | 2 | 2400 | 2 | 1110008(C | mobile | 4000 | 3 | 1110011 | bag | 800 |

Sheet1

READY

## 4] Recommendations:

Here, the customer id is stored in the first row product id of the products viewed by him are stored in the consecutive rows.



recommendation.xlsx - Excel

A9

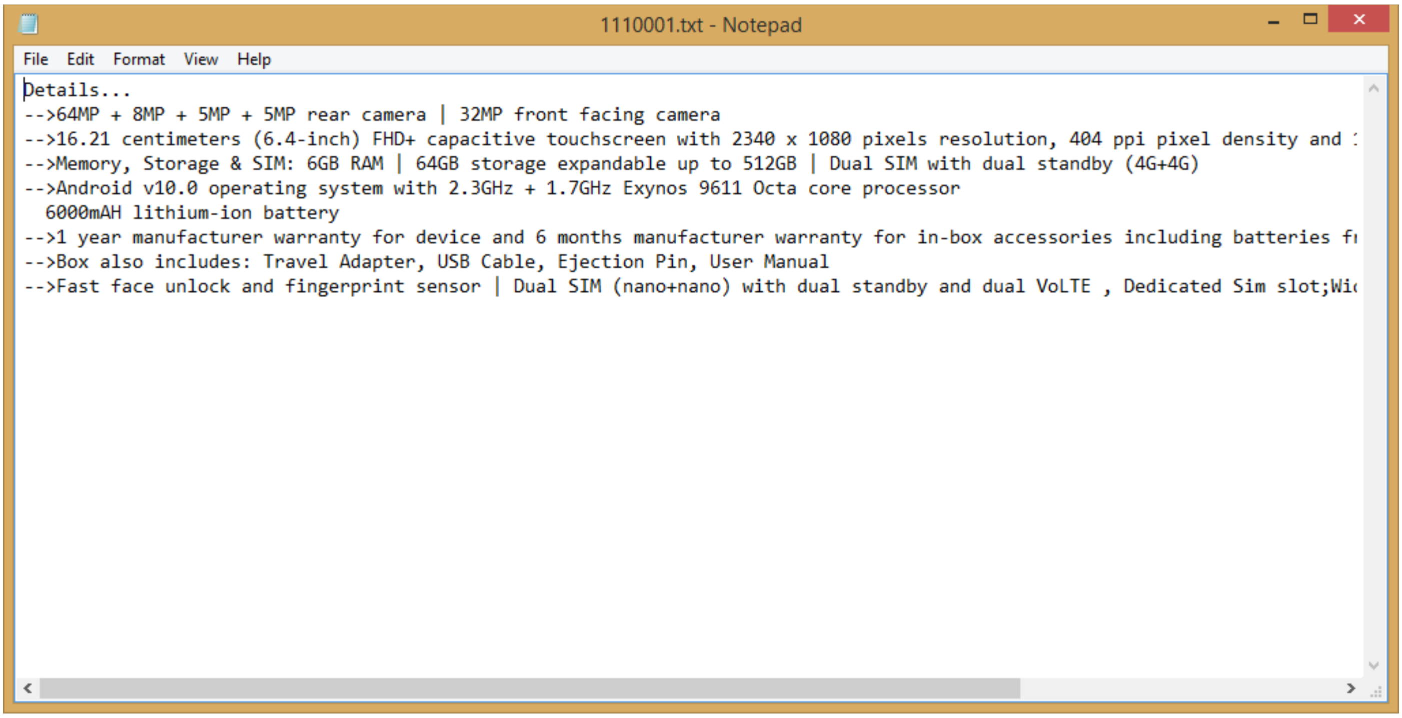
|  | A | B | C |
|---|---|---|---|
| 1 | 10000001 | 10000004 | 10000011 |
| 2 | 1110001 | 1110002 | 1110008 |
| 3 | 1110002 | 1110002 | 1110007 |
| 4 | 1110001 | 1110001 | 1110006 |
| 5 | 1110001 | 1110001 | 1110005 |
| 6 | 1110002 | 1110002 | 1110004 |
| 7 | 1110004 | 1110007 | 1110003 |
| 8 | 1110005 | 1110008 | 1110001 |

Sheet1

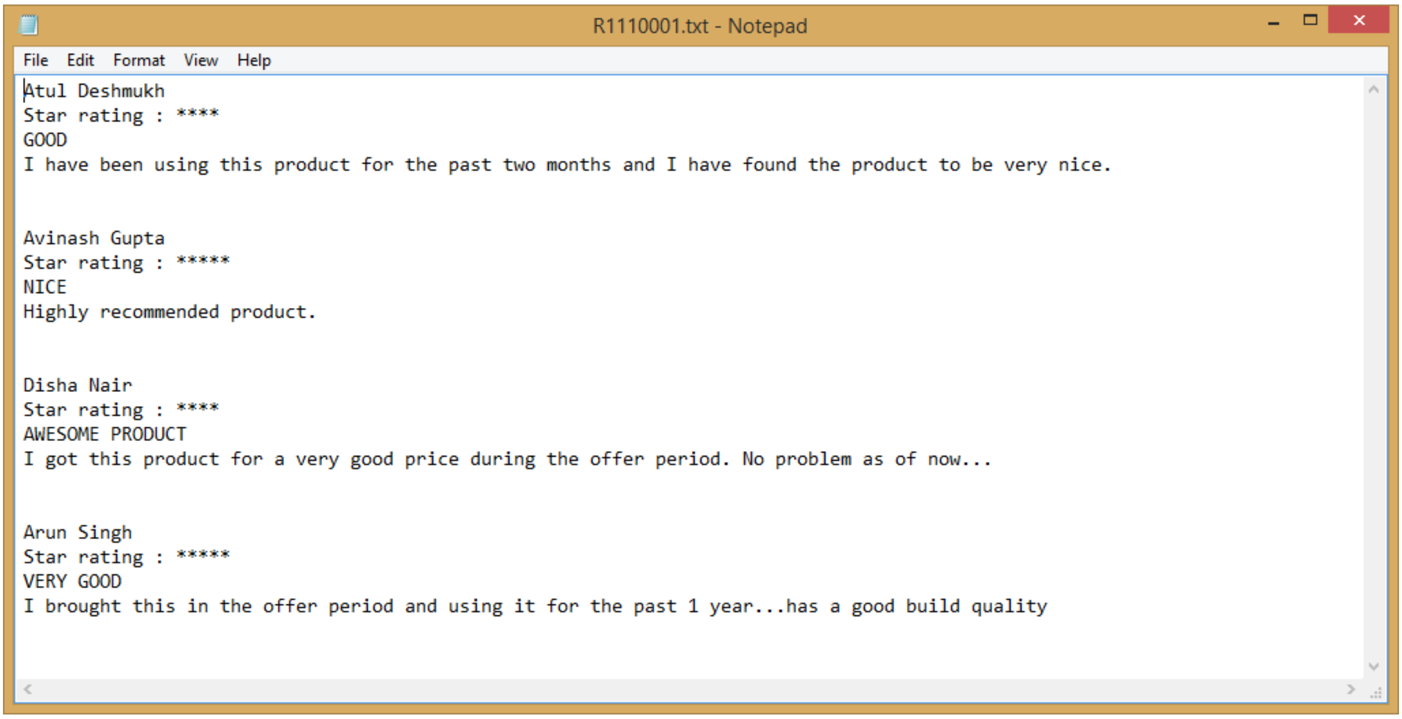
READY

## TEXT FILES:

Text files are used to view the product details and the review for a particular product.
Two files are used for each product. The first file has the detailed description for the product and is saved with the name of product id. The second file has the details of all the reviews and is stored with the name of product id using a prefix “R” .
The feedback entered by the user for a particular product are also added to the
review file.



```
Details...
-->64MP + 8MP + 5MP + 5MP rear camera | 32MP front facing camera
-->16.21 centimeters (6.4-inch) FHD+ capacitive touchscreen with 2340 x 1080 pixels resolution, 404 ppi pixel density and
-->Memory, Storage & SIM: 6GB RAM | 64GB storage expandable up to 512GB | Dual SIM with dual standby (4G+4G)
-->Android v10.0 operating system with 2.3GHz + 1.7GHz Exynos 9611 Octa core processor
  6000mAH lithium-ion battery
-->1 year manufacturer warranty for device and 6 months manufacturer warranty for in-box accessories including batteries f
-->Box also includes: Travel Adapter, USB Cable, Ejection Pin, User Manual
-->Fast face unlock and fingerprint sensor | Dual SIM (nano+nano) with dual standby and dual VoLTE , Dedicated Sim slot;Wi
```



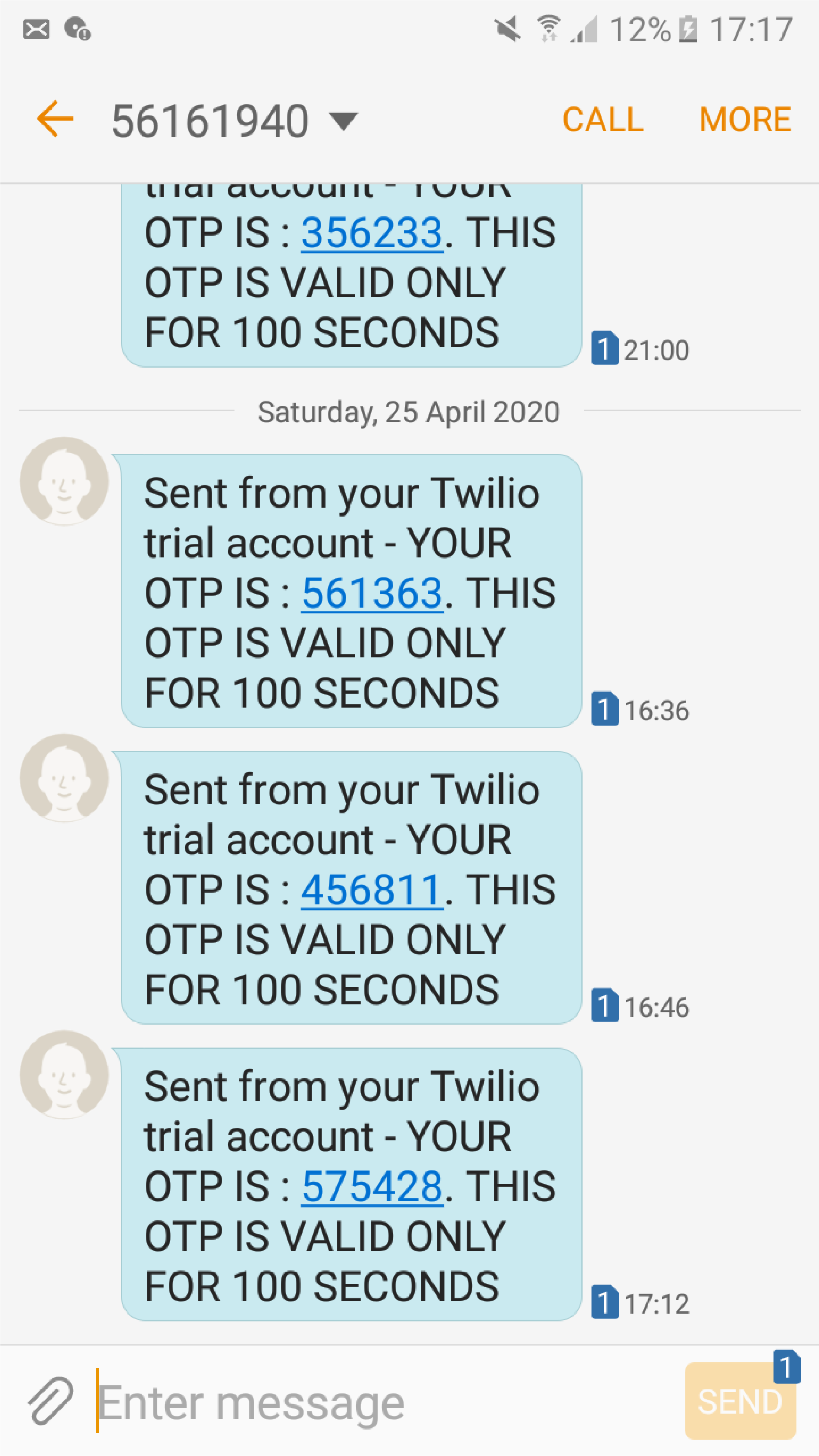
```
Atul Deshmukh
Star rating : ****
GOOD
I have been using this product for the past two months and I have found the product to be very nice.


Avinash Gupta
Star rating : *****
NICE
Highly recommended product.


Disha Nair
Star rating : ****
AWESOME PRODUCT
I got this product for a very good price during the offer period. No problem as of now...


Arun Singh
Star rating : *****
VERY GOOD
I brought this in the offer period and using it for the past 1 year...has a good build quality
```

**OTP SENT TO THE REGISTERED MOBILE NUMBER:**



trial account - YOUR OTP IS : 356233. THIS OTP IS VALID ONLY FOR 100 SECONDS

1 21:00

Saturday, 25 April 2020

Sent from your Twilio trial account - YOUR OTP IS : 561363. THIS OTP IS VALID ONLY FOR 100 SECONDS

1 16:36

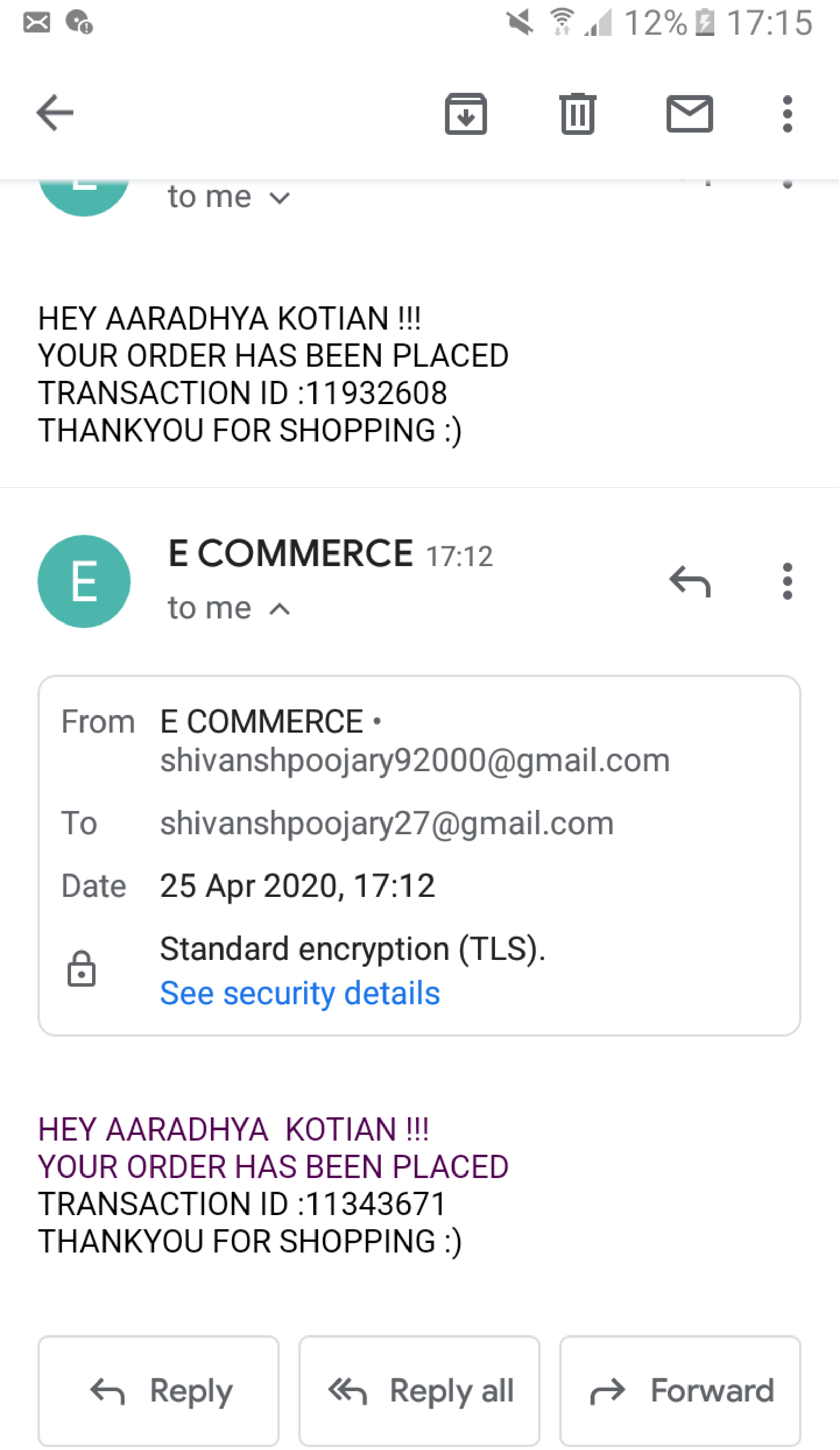
Sent from your Twilio trial account - YOUR OTP IS : 456811. THIS OTP IS VALID ONLY FOR 100 SECONDS

1 16:46

Sent from your Twilio trial account - YOUR OTP IS : 575428. THIS OTP IS VALID ONLY FOR 100 SECONDS

1 17:12

**CONFIRMATION SENT TO THE REGISTERED EMAIL ACCOUNT:**



to me

HEY AARADHYA KOTIAN !!!
YOUR ORDER HAS BEEN PLACED
TRANSACTION ID :11932608
THANKYOU FOR SHOPPING :)

E COMMERCE 17:12
to me

From E COMMERCE • shivanshpoojary92000@gmail.com

To shivanshpoojary27@gmail.com

Date 25 Apr 2020, 17:12

Standard encryption (TLS).
See security details

HEY AARADHYA  KOTIAN !!!
YOUR ORDER HAS BEEN PLACED
TRANSACTION ID :11343671
THANKYOU FOR SHOPPING :)

## SOURCE CODE :

```
d1=[]
cart=[]
from openpyxl import load_workbook
import openpyxl
def signup():
   from openpyxl import load_workbook
   import openpyxl
   wb=load_workbook("C:\\Users\\chaitra\\Desktop\\customer_details.xlsx")
   sheet=wb.worksheets[0]
   val=sheet.cell(row=1,column=20).value
   d2={}
   name=""
   name1=input("ENTER YOUR FIRST NAME:\t")
   name2=input("ENTER YOUR LAST NAME:\t")
   name=name1.capitalize()+" "+name2.capitalize()
   d2[1]=10000001+int(val)
   d2[2]=name
   dob=input("ENTER DATE OF BIRTH(dd/mm/yyyy):\t")
   d2[9]=dob
   while True:
      try:
         mob=int(input("ENTER MOBILE NUMBER:\t"))
      except ValueError:
         printf("WRONG INPUT !!...ENTER AGAIN")
      else:
         break
   d2[3]=mob
   g=input("ENTER GENDER\nM-->MALE\nF-->FEMALE\n")
   d2[10]=g
   d2[7]=0
   d2[8]=0
   em=input("ENTER E-MAIL ID\t:")
   d2[4]=em
   flag_u=1
   while flag_u==1:
      user_name=input("ENTER A USER NAME FOR FUTURE LOGIN:\t")
      wb=load_workbook("C:\\Users\\chaitra\\Desktop\\customer_details.xlsx")
      sheet=wb['Sheet1']
      for value in sheet.iter_cols(min_col=5,max_col=5,values_only=True):
         if user_name not in value:
            flag_u=0
            break
         else:
            print("THE ENTERED USER NAME ALREADY EXISTS IN OUR DATABASE...PLEASE ENTER THE USER
NAME AGAIN...")
   d2[5]=user_name
   while True:
        passcode=input("ENTER PASSCODE(min 8 characters):\t")
        if(len(passcode)<8):
            print("INVALID INPUT...PASSCODE SHOULD CONTAIN MINIMUM 8 CHARACTERS")
            continue
        else:
            break
    d2[6]=passcode
    d2[11]=0
    sheet.cell(row=1,column=20).value=str(int(val)+1)
    for i in range(1,12):
        sheet.cell(row=int(val)+2,column=i).value=d2.get(i)
    wb.save("C:\\Users\\chaitra\\Desktop\\customer_details.xlsx")
    global d1
    for i in range(2,int(sheet.cell(row=1,column=20).value)+2):
        if(user_name==sheet.cell(row=i,column=5).value):
            for element in range(1,12):
                d1.append(sheet.cell(row=i,column=element).value)
def find(user_name,passw):
    wb=load_workbook("C:\\Users\\chaitra\\Desktop\\customer_details.xlsx")
    sheet=wb['Sheet1']
    global d1
    for i in range(2,int(sheet.cell(row=1,column=20).value)+2):
        if(user_name==sheet.cell(row=i,column=5).value):
            for element in range(1,12):
                d1.append(sheet.cell(row=i,column=element).value)
    for value in sheet.iter_cols(min_col=5,max_col=5,values_only=True):
        if user_name in value:
            lfv1=value.index(user_name)
        else:
            return 0
    for value1 in sheet.iter_cols(min_col=6,max_col=6,values_only=True):
        if passw in value1:
            lfv2=value1.index(passw)
        else:
            return 0
    if(lfv1==lfv2):
        return 1
    else:
        return 0
def login():
    while True:
        try:
            choice1=int(input("ENTER THE CHOICE\n1-->LOGIN\n2-->SIGNUP\n"))
        except:
            print("\nINVALID INPUT...PLEASE TRY AGAIN")
        else:
            break
    while(choice1==1):
        user_name=input("ENTER YOUR USERNAME:\t")
        passw=input("ENTER YOUR PASSCODE:\t")
        r1=find(user_name,passw)
        if(r1==1):
            print("\nLOGIN SUCCESSFUL !!...")
            choice1=0
        while(r1!=1):
            print("\nINVALID USERNAME AND PASSCODE...PLEASE TRY AGAIN...\n")
            break
    if(choice1==2):
        signup()
login()
print(d1)
def twilio(n,o):
    from twilio.rest import Client
    main_b="YOUR OTP IS : "+str(o)+". THIS OTP IS VALID ONLY FOR 100 SECONDS"
    account_sid = 'AC7b0e6e37743976a12d90f73daeef1371'
    auth_token = '8b1640a6cdadf4cb042f9061eb262983'
    client = Client(account_sid, auth_token)

    message = client.messages                    .create(
                        body=main_b,
                        from_='+18122704351',
                        to='+91'+str(n)
                    )
def feedback(d1):
    from openpyxl import load_workbook
    import openpyxl
    fl,sv_id,c=0,[],6
    wb=load_workbook("C:\\Users\\chaitra\\Desktop\\transaction_details.xlsx")
    sheet=wb.worksheets[0]
    t_id=int(input("ENTER TRANSCATION ID SENT TO THE REGISTERED E-MAIL\n"))
    cell_val=sheet.cell(row=1,column=40).value
    while(fl!=1):
        for el in range(2,int(cell_val)+2):
            if(sheet.cell(row=el,column=1).value==t_id and sheet.cell(row=el,column=2).value==d1[0]):
                for j in range(sheet.cell(row=el,column=3).value):
                    print("\nNO. {0}\nPRODUCT ID = {1}\nPRODUCT NAME =
{2}".format(j+1,sheet.cell(row=el,column=c).value,sheet.cell(row=el,column=c+1).value))
                    sv_id.append(sheet.cell(row=el,column=c).value)
                    c+=4
                    fl=1
                break
        if(fl!=1):
            t_id=int(input("INVALID...ENTER AGAIN...OR ENTER 0 TO EXIT\n"))
            if(t_id==0):
                return
    p_inp=int(input("ENTER PRODUCT NUMBER\n"))
    a=""
    while True:
        try:
            num=int(input("ENTER YOUR RATING OUT OF FIVE\n"))
        except ValueError :
            print("WRONG INPUT !!!..\nPLEASE ENTER A NUMERIC VALUE OUT OF 5 ")
        else:
            break
    title=input("ENTER REVIEW TITLE:\n")
    det=input("ADD DETAILS...\n")
    print("TAHNKYOU FOR YOUR FEEDBACK..YOUR INPUT HAS BEEN SAVED..")
    a="\n\nName : "+d1[1].capitalize()+"\n"+"Star rating : "+"*"*num+"\n"+title.upper()+"\n" +det+"\n"
    str2="C:\\Users\\chaitra\\Desktop\\product_details\\"
    id2=".txt"
    fo2=open(str2+"R"+str(sv_id[p_inp-1])+id2,"a")
    fo2.write(a)
    fo2.close()
    from openpyxl import load_workbook
    import openpyxl
    wb=load_workbook("C:\\Users\\chaitra\\Desktop\\products_details.xlsx")
    sheet=wb.worksheets[0]
    c_val=sheet.cell(row=1,column=19).value
    for i in range(2,int(c_val)+2):
        if(sheet.cell(row=i,column=1).value==sv_id[p_inp-1]):
            #print("found")
            avg=float(sheet.cell(row=i,column=5).value)
            tot_n=int(sheet.cell(row=i,column=12).value)
            sheet.cell(row=i,column=5).value=round(((avg*tot_n)+num)/(tot_n+1),2)
            sheet.cell(row=i,column=12).value=tot_n+1
            wb.save("C:\\Users\\chaitra\\Desktop\\products_details.xlsx")
def check_valid_c(t_id,lim,ch):
    from openpyxl import load_workbook
    import openpyxl
    wb=load_workbook("C:\\Users\\chaitra\\Desktop\\transaction_details.xlsx")
    sheet=wb.worksheets[0]
    for m in range(2,lim+2):
        if(type(sheet.cell(row=m,column=1).value)==type(" ")):
            if(sheet.cell(row=m,column=1).value=="N"+str(t_id)):
                print("INVALID ACTION...THE TRANSACTION HAS ALREADY BEEN CANCELLED\nREQUEST IS DENIIED\n")
                return False
            else:
                if(sheet.cell(row=m,column=1).value)==str(ch)+"C"+str(t_id):
                    print("INVALID ACTION...THE PRODUCT HAS ALREADY BEEN CANCELLED\nREQUEST IS DENIED")
                    return False
    return True
def smtp1(name,e_id,t_id,amt,r=u"\u20B9"):
    str3="AN EMAIL REGARDING CONFIRMATION IS SENT TO REGISTERED EMAIL ID "+e_id
    str1="YOUR REQUEST FOR CANCELLATION IS ACCEPTED...REFUND OF AMOUNT "+r+" "+str(amt)+" WILL
BE INITIATED WITHIN 3 WORKING DAYS...THANK YOU"
    str2="HEY "+name+"...\n"+str1
    print(str1+"\n"+str3)
    import smtplib
    server=smtplib.SMTP('smtp.gmail.com',port=587)
    server.starttls()
    server.login("shivanshpoojary92000@gmail.com","ZYX54321lkj")
    msg=str2
    #print(msg)
    server.sendmail("shivanshpoojary92000@gmail.com",e_id,msg)
    server.quit()
def req_cancel(d1):
    from openpyxl import load_workbook
    import openpyxl
    fl,sv_id,c=0,[],6
    wb=load_workbook("C:\\Users\\chaitra\\Desktop\\transaction_details.xlsx")
    sheet=wb.worksheets[0]
    t_id=int(input("ENTER  TRANSCATION ID SENT TO THE REGISTERED E-MAIL\n"))
    cell_val=sheet.cell(row=1,column=40).value
    while(fl!=1):
        for el in range(2,int(cell_val)+2):
            if(sheet.cell(row=el,column=1).value==t_id and sheet.cell(row=el,column=2).value==d1[0]):
                sv_id.append(el)
                sv_id.append(sheet.cell(row=el,column=3).value)
                for j in range(sheet.cell(row=el,column=3).value):
                    print("\nNO.  {0}\nPRODUCT ID = {1}\nPRODUCT NAME =
{2}".format(j+1,sheet.cell(row=el,column=c).value,sheet.cell(row=el,column=c+1).value))
                    c+=4
                    fl=1
                break
        if(fl!=1):
            t_id=int(input("INVALID...ENTER  AGAIN...OR ENTER 0 TO EXIT\n"))
            if(t_id==0):
                return
    if(sv_id[1]==1):
        ch=int(input("ENTER 1 TO CONFIRM CANCELLATION OF THE ORDER\n"))
        if(ch==1 and check_valid_c(t_id,cell_val,ch)):
            sheet.cell(row=int(cell_val)+2,column=1).value="N"+str(t_id)
            sheet.cell(row=int(cell_val)+2,column=2).value=d1[0]
            sheet.cell(row=int(cell_val)+2,column=3).value=1
            for i in range(5):
                sheet.cell(row=int(cell_val)+2,column=i+4).value=sheet.cell(row=sv_id[0],column=i+4).value
            sheet.cell(row=1,column=40).value=int(cell_val)+1
            sheet.cell(row=sv_id[0],column=6+(ch-1)*4).value=str(sheet.cell(row=sv_id[0],column=6+(ch-
1)*4).value)+"(CANCELLED)"
            wb.save("C:\\Users\\chaitra\\Desktop\\transaction_details.xlsx")
            smtp1(d1[1],d1[3],"N"+str(t_id),sheet.cell(row=int(cell_val)+2,column=4).value)
```

```
      else:
         pass
         #call back
   else:
      ch=int(input("ENTER PRODUCT NUMBER TO CANCEL THE PRODUCT OR ENTER 0 TO CANCEL ENTIRE
TRANSACTION\n"))
      if(check_valid_c(t_id,cell_val,ch)):
         if(ch==0):
            cell_count=1
            while(sheet.cell(row=sv_id[0],column=cell_count).value):
               cell_count+=1
            for i in range(1,cell_count+1):
               if(i==3):
                  continue
               sheet.cell(row=int(cell_val)+2,column=i+1).value=sheet.cell(row=sv_id[0],column=i+1).value
            sheet.cell(row=int(cell_val)+2,column=1).value="N"+str(t_id)
            cell_count=6
            for i in range(sheet.cell(row=sv_id[0],column=3).value):

sheet.cell(row=sv_id[0],column=cell_count).value=str(sheet.cell(row=sv_id[0],column=cell_count).value)+"
(CANCELLED)"
               cell_count+=4
            sheet.cell(row=1,column=40).value=int(cell_val)+1
            k_l=0
            for i in range(2,int(cell_val)+2):
               if(type(sheet.cell(row=i,column=1).value)==type("  ") and
sheet.cell(row=i,column=2).value==d1[0]):
                  in_val=sheet.cell(row=i,column=1).value.find("C")
                  if(in_val!=-1):
                     new_id=sheet.cell(row=i,column=1).value[in_val+1:]
                     if(new_id==str(t_id)):
                        k_l+=sheet.cell(row=i,column=4).value
            sheet.cell(row=int(cell_val)+2,column=4).value=sheet.cell(row=sv_id[0],column=4).value-k_l
            wb.save("C:\\Users\\chaitra\\Desktop\\transaction_details.xlsx")
            smtp1(d1[1],d1[3],"N"+str(t_id),sheet.cell(row=int(cell_val)+2,column=4).value)
         else:
            dict1={2:2,(ch-1)*4+5:5,(ch-1)*4+6:6,(ch-1)*4+7:7,(ch-1)*4+8:8}
            for k,v in dict1.items():
               sheet.cell(row=int(cell_val)+2,column=v).value=sheet.cell(row=sv_id[0],column=k).value
               if(k==(ch-1)*4+6):

sheet.cell(row=sv_id[0],column=k).value=str(sheet.cell(row=sv_id[0],column=k).value)+"(CANCELLED)"
               sheet.cell(row=int(cell_val)+2,column=3).value=1
            sheet.cell(row=int(cell_val)+2,column=1).value=str(ch)+"C"+str(t_id)

sheet.cell(row=int(cell_val)+2,column=4).value=sheet.cell(row=sv_id[0],column=(ch*4)+4).value*sheet.cell
(row=sv_id[0],column=(ch*4)+1).value
            sheet.cell(row=1,column=40).value=int(cell_val)+1
            wb.save("C:\\Users\\chaitra\\Desktop\\transaction_details.xlsx")
            smtp1(d1[1],d1[3],str(ch)+"C"+str(t_id),sheet.cell(row=int(cell_val)+2,column=4).value)

def email(e,name,id1):
   import smtplib
   from email.message import EmailMessage
   str1="HEY "+name.upper()+"  !!!\nYOUR ORDER HAS BEEN PLACED\nTRANSACTION ID
:"+id1+"\nTHANKYOU  FOR SHOPPING :)\n"
   email=EmailMessage()
   email['from']="E COMMERCE"
   email['to']=e
   email['subject']="ORDER CONFIRMATION"
   email.set_content(str1)
   with smtplib.SMTP(host="smtp.gmail.com",port=587)  as smtp:
      smtp.ehlo()
      smtp.starttls()
      smtp.login("shivanshpoojary92000@gmail.com","ZYX54321lkj")
      smtp.send_message(email)
def otp():
   import numpy as np
   result=''
   for i in range(6):
      a=np.random.randint(1,9,size=(1,1))
      a1=a[0,0]
      a2=str(a1)
      result=result+a2
   return result

def bill(d1,cart):
   from openpyxl import load_workbook
   import openpyxl
   wb=load_workbook("C:\\Users\\chaitra\\Desktop\\products_details.xlsx")
   sheet=wb.worksheets[0]
   c_val=sheet.cell(row=1,column=19).value
   prod_id,prod_name,prod_price,prod_q,price,prod_save=[],[],[],[],[],[]
   total_sav,total=0,0
   dict1={}
   r=u"\u20B9"
   import pandas as pd
   from datetime import date
   from datetime import datetime
   now=datetime.now()
   c_t=now.strftime("%H:%M:%S")
   today=date.today()
   c_d=today.strftime("%B  %d, %Y")
   t_id=int(d1[0])+int(cart[1][0])+int(otp())
   print("\t\t\t\t***  INVOICE ***\n")
```

```
    print("NAME  : {0}\nCUSTOMER ID : {1}\nDATE : {2}\nTIME  :{3}\nTRANSACTION ID
:{4}\n".format(d1[1],d1[0],c_d,c_t,t_id))
    for i in range(0,len(cart),2):
        prod_q.append(cart[i])
        prod_id.append(cart[i+1][0])
        prod_price.append(cart[i+1][1])
        price.append((cart[i])*(cart[i+1][1]))
        prod_save.append((cart[i+1][5])*cart[i])
        prod_name.append(cart[i+1][6])
        total_sav+=cart[i+1][5]*cart[i]
        total+=cart[i]*cart[i+1][1]
    curr_st=0
    for i in range(0,len(cart),2):
        for j in range(2,int(c_val)+2):
            if(sheet.cell(row=j,column=1).value==cart[i+1][0]):
                sheet.cell(row=j,column=10).value-=cart[i]
                if(sheet.cell(row=j,column=10).value<=0):
                    if(sheet.cell(row=j,column=10).value==0):
                        print("WE  ARE SORRY..CURRENTLY THE PRODUCT IS OUT OF STOCK...")
                      #call back
                    else:
                        inp=int(input("ENTER  CHOICE\n1-->LIMITED STOCK AVAILABLE...PROCEED TO BUY WITH {0}
QUANTITIES\n2-->CANCEL OEDER AND GO  BACK\n".format(sheet.cell(row=j,column=10).value+cart[i])))
                        if(inp==2):
                            sheet.cell(row=j,column=10).value+=cart[i]
                            pass#call back
                        sheet.cell(row=j,column=10).value=0
                wb.save("C:\\Users\\chaitra\\Desktop\\products_details.xlsx")
    dict1={'PRODUCT NAME':prod_name,'PRODUCT  ID':prod_id,"PRODUCT
PRICE":prod_price,"QUANTITY":prod_q,"TOTAL":price,"YOU  SAVE":prod_save}
    df=pd.DataFrame(dict1)
    print(df)
    print("TOTAL  AMOUNT:{0}{1}\nTOTAL  SAVINGS:{2}{3}".format(r,total,r,total_sav))
    final1=int(input("ENTER  1 TO PLACE YOUR ORDER AND PROCEED TO PAY...\n"))
    if(final1==1):
        wb=load_workbook("C:\\Users\\chaitra\\Desktop\\customer_details.xlsx")
        sheet=wb.worksheets[0]
        c_val=sheet.cell(row=1,column=20).value
        for i in range(2,int(c_val)+2):
            #print(sheet.cell(row=i,column=1).value)
            if(sheet.cell(row=i,column=1).value==d1[0]):
                if(sheet.cell(row=i,column=7).value):
                    s1=sheet.cell(row=i,column=7).value
                    cvv=sheet.cell(row=i,column=8).value
                    s1="*"*4+" "+"*"*4+" "+"*"*4+" "+str(s1)[12::]
                    print("DEBIT  CARD NUMBER  :{}".format(s1))
                    cvv2=int(input("ENTER  THE CVV\t"))
                    while(cvv2!=int(cvv)):
```

```
                    cvv2=int(input("INVALID...ENTER  THE CVV AGAIN..\t"))
                break
            else:
                s1=input("ENTER 16 DIGIT DEBIT CARD NUMBER\n")
                cvv=input("ENTER THE CVV\n")
                u_choice=int(input("SAVE  YOUR DEBIT CARD DETAILS FOR FUTURE TRANSACTIONS ?\nYES--
>1\nNO-->2\n"))
                if(u_choice==1):
                    sheet.cell(row=i,column=7).value=s1
                    sheet.cell(row=i,column=8).value=cvv
                    wb.save("C:\\Users\\chaitra\\Desktop\\customer_details.xlsx")
                break
    print("ACCEPTED\nAN OTP HAS BEEN SENT TO THE REGISTERED MOBILE NUMBER : {0}".format(d1[2]))
    otp1=otp()
    twilio(d1[2],otp1)
    #print(otp1)
    import time
    otp_count=2
    t1=time.time()
    otp2=int(input("ENTER  THE OTP:\t"))
    while(otp_count!=0 and otp2!=int(otp1)):
        otp2=int(input("ENTER  THE OTP AGAIN:\t"))
        otp_count-=1
    t2=time.time()
    if(t2-t1>100 or otp_count==0):
        print("TRANSACTION  FAILED\n")
        bill(d1,cart)
    else:
        print("OTP ACCEPTED !!..\nTRANSACTION SUCCESSFUL\n")
        wb=load_workbook("C:\\Users\\chaitra\\Desktop\\transaction_details.xlsx")
        sheet=wb.worksheets[0]
        val_t=sheet.cell(row=1,column=40).value
        sheet.cell(row=int(val_t)+2,column=1).value=t_id
        sheet.cell(row=int(val_t)+2,column=2).value=d1[0]
        sheet.cell(row=int(val_t)+2,column=3).value=len(cart)/2
        sheet.cell(row=int(val_t)+2,column=4).value=total
        c1=4
        for i in range(1,len(cart)+1):
            if(i%2==1):
                sheet.cell(row=int(val_t)+2,column=i+c1).value=cart[c1-4]
            else:
                sheet.cell(row=int(val_t)+2,column=i+c1).value=cart[c1-3][0]
                c1+=1
                sheet.cell(row=int(val_t)+2,column=i+c1).value=cart[c1-4][6]
                c1+=1
                sheet.cell(row=int(val_t)+2,column=i+c1).value=cart[c1-5][1]
        sheet.cell(row=1,column=40).value=int(val_t)+1
        wb.save("C:\\Users\\chaitra\\Desktop\\transaction_details.xlsx")
```

```
            print("PROCESSING  YOUR ORDER REQUEST...\n")
            email(str(d1[3]),str(d1[1]),str(t_id))
            time.sleep(10)
            print("ORDER IS PLACED !!!..\nORDER CONFIRMATION WITH TRANSACTION ID IS SENT TO THE
REGISTERED G-MAIL ID 1: {0}\nTHANKYOU  FOR SHOPPING  WITH US".format(d1[3]))
#d1=[10000011,'RAJESH
KOLI',8291500733,'abc@gmail.com','rajesh_k','12345678',0,0,'23/06/2000','M','4589']

def display_sorted(new_l,r=u"\u20B9"):
   from openpyxl import load_workbook
   import openpyxl
   wb=load_workbook("C:\\Users\\chaitra\\Desktop\\products_details.xlsx")
   sheet=wb.worksheets[0]
   #print(new_l)
   for i in range(len(new_l)):
      print("{0}\n{4}{5}\nRating  : {1}\nYOU SAVE
:{6}{2}({3}%)\n".format(new_l[i][6],new_l[i][2],new_l[i][5],new_l[i][4],r,new_l[i][1],r))


def filter_products(new_l):
   choice_app=int(input("SELECT  THE REQUIRED FILTER\n1-->LOW TO HIGH(PRICE)\n2-->HIGH TO LOW\n3-
->BASED ON RATING\n"))
   if(choice_app==1):
      new_l.sort(key=lambda  x:x[1])
      #print(new_l)
      display_sorted(new_l)
   elif(choice_app==2):
      new_l.sort(key=lambda  x:x[1],reverse=True)
      display_sorted(new_l)
   elif(choice_app==3):
      new_l.sort(key=lambda  x:x[2],reverse=True)
      display_sorted(new_l)
   else:
      print("INVALID  INPUT !!!...")
      filter_products(new_l)




def fetch(product_id):
   global d1
   count_p,save=0,0
   list_p,list_pr,list_r,list_i,list_percent,list_save,list_n,list_rn=[],[],[],[],[],[],[],[]
   per=0.0
   r=u"\u20B9"
   from openpyxl import load_workbook
   import openpyxl
   wb=load_workbook("C:\\Users\\chaitra\\Desktop\\products_details.xlsx")
   sheet=wb.worksheets[0]
```

```
    parent_id=product_id
    val=sheet.cell(row=1,column=19).value
    for i in range(2,int(val)+1):
        if(int(sheet.cell(row=i,column=2).value)==parent_id):
            count_p+=1
            list_i.append(i)
            list_n.append("PRODUCT {0}".format(count_p))
            list_rn.append(sheet.cell(row=i,column=12).value)
            list_p.append(sheet.cell(row=i,column=1).value)
            list_pr.append(sheet.cell(row=i,column=4).value)
            list_r.append(sheet.cell(row=i,column=5).value)
            per=round((100-(sheet.cell(row=i,column=4).value/sheet.cell(row=i,column=3).value)*100),2)
            save=sheet.cell(row=i,column=3).value-sheet.cell(row=i,column=4).value
            list_percent.append(per)
            list_save.append(save)
            print("{0}\n{4}{5}\nRating  : {1}\nYOU SAVE :{6}{2}({3}%)\n".format(list_n[count_p-
1],sheet.cell(row=i,column=5).value,save,per,r,sheet.cell(row=i,column=4).value,r))
    new_l=list(zip(list_p,list_pr,list_r,list_i,list_percent,list_save,list_n,list_rn))
    det=int(input("ENTER  1 TO APPLY FILTERS OR 0 TO SKIP THIS STEP :\n"))
    if(det==1):
        #new_l=list(zip(list_p,list_pr,list_r,list_i,list_percent,list_save,list_n,list_rn))
        filter_products(new_l)
    det1=int(input("ENTER  PRODUCT NUMBER TO GET MORE DETAILS OR O TO SKIP\t"))
    if(det1!=0):
        if(det1<=count_p):
            product_id+=det1
            str1="C:\\Users\\chaitra\\Desktop\\product_details\\"
            id1=".txt"
            fo=open(str1+str(product_id)+id1,"r")
            data=fo.read()
            print(data)
        else:
            print("INVALID  INPUT")
            head_pg()
    for z in range(len(new_l)):
        if(product_id==int(new_l[z][0])):
            tup=new_l[z]
    ch=int(input("ENTER  1 TO VIEW REVIEWS...\t"))
    if(ch==1):
        print("TOTAL NUMBER  OF REVIEWS={0}\nAVERAGE  RATING={1}".format(tup[7],tup[2]))
        str2="C:\\Users\\chaitra\\Desktop\\product_details\\"
        id2=".txt"
        fo2=open(str2+"R"+str(product_id)+id2,"r")
        data2=fo2.read()
        print(data2)
    choice_checkout=int(input("ENTER  THE CHOICE\n1-->BUY NOW\n2-->ADD TO CART\n3-->GO BACK\n"))
    for z in range(len(new_l)):
        if(product_id==int(new_l[z][0])):
            tup=new_l[z]
    if(choice_checkout==3):
        d1=[10000011]
        a,st=1,0
        wb1=load_workbook("C:\\Users\\chaitra\\Desktop\\recommendation.xlsx")
        sheet1=wb1.worksheets[0]
        while(sheet1.cell(row=1,column=a).value):
            if(sheet1.cell(row=1,column=a).value==d1[0]):
                st=a
                break
            a+=1
        if(st==0):
            sheet1.cell(row=1,column=a).value=d1[0]
            st=a
        a=2
        while(sheet1.cell(row=a,column=st).value):
            a+=1
        sheet1.cell(row=a,column=st).value=product_id
        wb1.save("C:\\Users\\chaitra\\Desktop\\recommendation.xlsx")
        head_pg()
    elif(choice_checkout==1):
        quant=int(input("ENTER  THE QUANTITY:\t"))
        bill(d1,[quant,tup])
    elif(choice_checkout==2):
        quant=int(input("ENTER  THE QUANTITY:\t"))
        global cart
        cart.extend([quant,tup])
        checkout_2=input("ENTER  C TO CONTINUE SHOPPING OR P TO CHECKOUT...")
        if(checkout_2=="C"):
            head_pg()
        else:
            bill(d1,cart)

    else:
        print("INVALID  INPUT...\n")
        head_pg()
def check_my_rec(cust_id):
    prod_id=0
    wb=load_workbook("C:\\Users\\chaitra\\Desktop\\transaction_details.xlsx")
    sheet=wb.worksheets[0]
    u=sheet.cell(row=1,column=40).value
    for i in range(int(u),1,-1):
        if(sheet.cell(row=i,column=2).value==cust_id):
            if(type(sheet.cell(row=i,column=1).value)==type("  ")):
                continue
            else:
                prod_id=sheet.cell(row=i,column=6).value
    prod_id=(prod_id//10000)*10000
    wb1=load_workbook("C:\\Users\\chaitra\\Desktop\\recommendation.xlsx")
    sheet1=wb1.worksheets[0]
    a,b,l1,l2=1,1,[],[]
    while(sheet1.cell(row=1,column=a).value):
        if(cust_id==sheet1.cell(row=1,column=a).value):
            while(sheet1.cell(row=b+1,column=a).value):
                l1.append(sheet1.cell(row=b+1,column=a).value)
                b+=1
        a+=1
    a=1
    while(sheet1.cell(row=1,column=a).value):
        b=2
        while(sheet1.cell(row=b,column=a).value):
            l2.append(sheet1.cell(row=b,column=a).value)
            b+=1
        a+=1
    b=max(set(l2),key=l2.count)
    b=(b//10000)*10000
    if(l1):
        a=max(set(l1),key=l1.count)
        a=(a//10000)*10000
        fetch(a)
    elif(prod_id):
        fetch(prod_id)
    else:
        b=max(set(l2),key=l2.count)
        b=(b//10000)*10000
        fetch(b)
def shop_now():
    product_id=1000000
    shop1=int(input("SELECT CATEGORY :\n\n1--> Mobiles, Computers\n2--> TV, Appliances, Electronics\n3-
-> Men\'s Fashion\n4--> Women\'s Fashion\n5--> Home, Kitchen, Pets, Furniture\n6--> Grocery and
Household appliances\n7--> Sports, Fitness, Bags, Luggage\n8--> Toy\'s, Kid\'s Fashion\n9--> Car,
Motorbike, Industrial\n10--> Books and Audible\n"))
    if(shop1==1):
        shop_sub1=int(input("SELECT SUB-CATEGORY :\n1--> Mobiles\n2--> Mobile Accessories\n3--> Power
Banks\n4--> Laptops\n5--> Tablets\n6--> Computers and Accessories\n7--> Office Supplies &
Stationary\n"))
        product_id+=(100000+shop_sub1*10000)
        fetch(product_id)
    elif(shop1==2):
        shop_sub1=int(input("SELECT SUB-CATEGORY :\n1-->Televisions\n2-->Washing machines\n3--
>Refrigerators\n4-->Microwaves\n5-->Air Conditioners\n6-->Dishwashers\n7-->Home Entertainmentt
systems"))
        product_id+=(200000+shop_sub1*10000)
        fetch(product_id)
    elif(shop1==3):
        shop_sub1=int(input("SELECT SUB-CATEGORY :\n1-->Clothing\n2-->Shoes\n3-->Watches\n4-->Bags
and Wallets\n5-->Sunglasses\n6-->Sportswear\n"))
        product_id+=(300000+shop_sub1*10000)
        fetch(product_id)
    elif(shop1==4):
        shop_sub1=int(input("SELECT SUB-CATEGORY :\n1-->Clothing\n2-->Shoes\n3-->Watches\n4-->Fashion
and Silver Jewellery\n5-->Gold and Diamond Jewellery\n6-->Handbags and Clutches\n7-->Sunglasses\n8--
>Sportswear\n"))
        product_id+=(400000+shop_sub1*10000)
        fetch(product_id)
    elif(shop1==5):
        shop_sub1=int(input("SELECT SUB-CETEGORY :\n1-->Kitchen and Dining\n2-->Home Furnishing\n3--
>Home Decor\n4-->Garden and Outdoors\n5-->Indoor Lighting\n6-->Pet Supplies\n7-->Art and Craft
Supplies"))
        product_id+=(500000+shop_sub1*10000)
        fetch(product_id)
    elif(shop1==6):
        shop_sub1=int(input("SELECT SUB-CATEGORY :\n1-->Super Saver Pantry\n2-->Cooking Essentials\n3--
>Snacks and Beverages\n4-->Packaged Foods\n5-->Gourmet Foods\n"))
        product_id+=(600000+shop_sub1*10000)
        fetch(product_id)
    elif(shop1==7):
        shop_sub1=int(input("SELECT SUB-CATEGORY :\n1-->Exercise and Fitness\n2-->Cycling\n3--
>Badminton\n4-->Cricket\n5-->Football\n6-->Other games\n7-->Backpacks\n8-->Travel Luggage"))
        product_id+=(700000+shop_sub1*10000)
        fetch(product_id)
    elif(shop1==8):
        shop_sub1=int(input("ENTER SUB-CATEGORY :\n1-->All Toys and Games\n2-->Kid\'s Clothing\n3--
>Kid\'s Shoes\n4-->School Bags\n"))
        product_id+=(800000+shop_sub1*10000)
        fetch(product_id)
    elif(shop1==9):
        shop_sub1=int(input("ENTER SUB-CATEGORY :\n1-->Motorbike Accessories and Parts\n2-->Car
Accessories\n3-->Car and Motorbike Care\n4-->Industrial and Scientific Supplies\n5-->Lab and
Scientific\n"))
        product_id+=(900000+shop_sub1*10000)
        fetch(product_id)
    elif(shop1==10):
        shop_sub1=int(input("ENTER SUB-CATEGORY :\n1-->Fiction Books\n2-->Comic books\n3-->School
Textbooks\n4-->Exam Central\n5-->Textbooks\n6-->Indian Language Books\n"))
        product_id+=(1000000+shop_sub1*10000)
        fetch(product_id)
    else:
        print("INVALID INPUT !!!...")
        shop_now()
def head_pg():
    choice_m=int(input("ENTER YOUR CHOICE...\n1-->CHECK MY RECOMMENDATIONS\n2-->SHOP NOW\n3-
->REQUEST CANCELLATION\n4-->POST A REVIEW\n5-->ADD MONEY TO E-WALLET\n"))
```

```
    global d1
    if(choice_m==1):
        check_my_rec(d1[0])
    elif(choice_m==2):
        shop_now()
    elif(choice_m==3):
        req_cancel(d1)
    elif(choice_m==4):
        feedback(d1)
    elif(choice_m==5):
        e_wallet()
    else:
        print("INVALID INPUT...PLEASE TRY AGAIN...")
        head_pg()

while True:
    try:
        head_pg()
    except ValueError:
        print("OOPS !!...SOMETHING WENT WRONG..TRY AGAIN... ")
    else:
        break
```

# OUTPUT:

```
runfile('C:/Users/chaitra/Downloads/Untitled31.py', wdir='C:/Users/chaitra/Downloads')

ENTER THE CHOICE
1-->LOGIN
2-->SIGNUP
2

ENTER YOUR FIRST NAME:      aaradhya

ENTER YOUR LAST NAME:       kotian

ENTER DATE OF BIRTH(dd/mm/yyyy):            20/10/2000

ENTER MOBILE NUMBER:        8291500733

ENTER GENDER
M-->MALE
F-->FEMALE
F

ENTER E-MAIL ID         :shivanshpoojary27@gmail.com

ENTER A USER NAME FOR FUTURE LOGIN:    aaradhya_k21

ENTER PASSCODE(min 8 characters): 8765428973

ENTER YOUR CHOICE...
1-->CHECK MY RECOMMENDATIONS
2-->SHOP NOW
3-->REQUEST CANCELLATION
4-->POST A REVIEW
5-->ADD MONEY TO E-WALLET
2

SELECT CATEGORY :

1--> Mobiles, Computers
2--> TV, Appliances, Electronics
3--> Men's Fashion
4--> Women's Fashion
5--> Home, Kitchen, Pets, Furniture
6--> Grocery and Household appliances
7--> Sports, Fitness, Bags, Luggage
8--> Toy's, Kid's Fashion
9--> Car, Motorbike, Industrial
```

```
10--> Books and Audible
1

SELECT SUB-CATEGORY :
1--> Mobiles
2--> Mobile Accessories
3--> Power Banks
4--> Laptops
5--> Tablets
6--> Computers and Accessories
7--> Office Supplies & Stationary
1
PRODUCT 1
₹12000
Rating : 4.5
YOU SAVE :₹3000(20.0%)

PRODUCT 2
₹18000
Rating : 4
YOU SAVE :₹2000(10.0%)

PRODUCT 3
₹9500
Rating : 3.5
YOU SAVE :₹500(5.0%)

PRODUCT 4
₹15000
Rating : 4.7
YOU SAVE :₹1000(6.25%)

PRODUCT 5
₹11000
Rating : 4.9
YOU SAVE :₹1000(8.33%)

PRODUCT 6
₹5800
Rating : 3
YOU SAVE :₹200(3.33%)

PRODUCT 7
₹38000
Rating : 3.7
YOU SAVE :₹2000(5.0%)

PRODUCT 8
```

```
₹9500
Rating : 4.16
YOU SAVE :₹500(5.0%)

PRODUCT 9
₹15000
Rating : 4.4
YOU SAVE :₹1000(6.25%)

PRODUCT 10
₹11000
Rating : 3.5
YOU SAVE :₹1000(8.33%)

PRODUCT 11
₹15000
Rating : 4.7
YOU SAVE :₹1000(6.25%)

PRODUCT 12
₹11000
Rating : 4.9
YOU SAVE :₹1000(8.33%)

PRODUCT 13
₹12000
Rating : 4.5
YOU SAVE :₹3000(20.0%)

PRODUCT 14
₹18000
Rating : 4
YOU SAVE :₹2000(10.0%)

PRODUCT 15
₹15000
Rating : 3.5
YOU SAVE :₹1000(6.25%)

PRODUCT 16
₹11000
Rating : 4.7
YOU SAVE :₹1000(8.33%)


ENTER 1 TO APPLY FILTERS OR 0 TO SKIP THIS STEP :
1
```

```
SELECT THE REQUIRED FILTER
1-->LOW TO HIGH(PRICE)
2-->HIGH TO LOW
3-->BASED ON RATING
3
PRODUCT 5
₹11000
Rating : 4.9
YOU SAVE :₹1000(8.33%)

PRODUCT 12
₹11000
Rating : 4.9
YOU SAVE :₹1000(8.33%)

PRODUCT 4
₹15000
Rating : 4.7
YOU SAVE :₹1000(6.25%)

PRODUCT 11
₹15000
Rating : 4.7
YOU SAVE :₹1000(6.25%)

PRODUCT 16
₹11000
Rating : 4.7
YOU SAVE :₹1000(8.33%)

PRODUCT 1
₹12000
Rating : 4.5
YOU SAVE :₹3000(20.0%)

PRODUCT 13
₹12000
Rating : 4.5
YOU SAVE :₹3000(20.0%)

PRODUCT 9
₹15000
Rating : 4.4
YOU SAVE :₹1000(6.25%)

PRODUCT 8
₹9500
Rating : 4.16
```

```
YOU SAVE :₹500(5.0%)

PRODUCT 2
₹18000
Rating : 4
YOU SAVE :₹2000(10.0%)

PRODUCT 14
₹18000
Rating : 4
YOU SAVE :₹2000(10.0%)

PRODUCT 7
₹38000
Rating : 3.7
YOU SAVE :₹2000(5.0%)

PRODUCT 3
₹9500
Rating : 3.5
YOU SAVE :₹500(5.0%)

PRODUCT 10
₹11000
Rating : 3.5
YOU SAVE :₹1000(8.33%)

PRODUCT 15
₹15000
Rating : 3.5
YOU SAVE :₹1000(6.25%)

PRODUCT 6
₹5800
Rating : 3
YOU SAVE :₹200(3.33%)


ENTER PRODUCT NUMBER TO GET MORE DETAILS OR O TO SKIP    1
Details...
-->64MP + 8MP + 5MP + 5MP rear camera | 32MP front facing camera
-->16.21 centimeters (6.4-inch) FHD+ capacitive touchscreen with 2340 x 1080 pixels resolution, 404 ppi
pixel density and 16M color support
-->Memory, Storage & SIM: 6GB RAM | 64GB storage expandable up to 512GB | Dual SIM with dual
standby (4G+4G)
-->Android v10.0 operating system with 2.3GHz + 1.7GHz Exynos 9611 Octa core processor
  6000mAH lithium-ion battery
```

```
-->1 year manufacturer warranty for device and 6 months manufacturer warranty for in-box accessories
including batteries from the date of purchase
-->Box also includes: Travel Adapter, USB Cable, Ejection Pin, User Manual
-->Fast face unlock and fingerprint sensor | Dual SIM (nano+nano) with dual standby and dual VoLTE ,
Dedicated Sim slot;Widevine L1 certification for HD streaming


ENTER 1 TO VIEW REVIEWS... 1
TOTAL NUMBER OF REVIEWS=4
AVERAGE RATING=4.5
Atul Deshmukh
Star rating : ****
GOOD
I have been using this product for the past two months and I have found the product to be very nice.


Avinash Gupta
Star rating : *****
NICE
Highly recommended  product.


Disha Nair
Star rating : ****
AWESOME PRODUCT
I got this product for a very good price during the offer period. No problem as of now...


Arun Singh
Star rating : *****
VERY GOOD
I brought this in the offer period and using it for the past 1 year...has a good build quality


ENTER THE CHOICE
1-->BUY NOW
2-->ADD TO CART
3-->GO BACK
1

ENTER THE QUANTITY:           2
                       *** INVOICE ***

NAME : Aaradhya Kotian
CUSTOMER ID : 10000030
DATE : April 25, 2020
TIME :19:47:13
TRANSACTION ID :11733512
```

```
  PRODUCT NAME  PRODUCT ID  PRODUCT PRICE  QUANTITY  TOTAL  YOU SAVE
0       PRODUCT 1    1110001         12000              2      24000     6000
TOTAL AMOUNT:₹24000
TOTAL SAVINGS:₹6000

ENTER 1 TO PLACE YOUR ORDER AND PROCEED TO PAY...
1

ENTER 16 DIGIT DEBIT CARD NUMBER
2234765409873453

ENTER THE CVV
212

SAVE YOUR DEBIT CARD DETAILS FOR FUTURE TRANSACTIONS ?
YES-->1
NO-->2
1
ACCEPTED
AN OTP HAS BEEN SENT TO THE REGISTERED MOBILE NUMBER : 8291500733
ENTER THE OTP:        731433
OTP ACCEPTED !!..
TRANSACTION SUCCESSFUL

PROCESSING YOUR ORDER REQUEST...

ORDER IS PLACED !!!..
ORDER CONFIRMATION WITH TRANSACTION ID IS SENT TO THE REGISTERED G-MAIL ID 1:
shivanshpoojary27@gmail.com
THANKYOU FOR SHOPPING WITH US
```

## REFERENCES:

- ‘Introduction to computing and problem solving using python’ by E Balagurusamy
- geeksforgeeks.org
- realpython.com
- tutorialspoint.com
- stackoverflow.com
- quora.com
- w3schools.com
- learnpython.org
- programiz.com