MAR THOMA PUBLIC SCHOOL
KAKKANAD

COMPUTER SCIENCE
PROJECT REPORT
CLASS XI
2023-24

Group Leader: **Aaron Shenny**

Group Members: **Aswin Aravind, Ardith R. , Aaron Jimmy, Rishab Binod**

CERTIFICATE

Certified that this is the bonafide record of the Project

Work of _____

_____

Of Class XI, MAR THOMA PUBLIC SCHOOL,

Submitted for COMPUTER SCIENCE Practical

Examination held on _____during academic

year 2023-2024.

Internal Examiner Principal Teacher-in-charge


SHOPIFY
Freshness Delivered, Shop 'til You Crop

ACKNOWLEDGEMENT

It is with great honor and gratitude that we extend our heartfelt appreciation to those whose unwavering support, guidance, and expertise have been instrumental in the completion of this project : SHOPIFY - AN ONLINE GROCERY SHOP. Through countless hours of dedication and hard work, our team has navigated challenges, celebrated victories, and embraced the spirit of collaboration. This project stands as a testament to the collective efforts of each member, showcasing the power of teamwork and synergy. We appreciate each other's contribution and are grateful to our Computer Science teacher Mrs. Reeba John who taught us the python programming language for the past year and with whose guidance we were able to make this project a complete success and our Principal Dr. Sheela Seth, for giving us a golden opportunity to do this project. In closing, we extend our deepest gratitude to all those involved, directly or indirectly, in this project's realization. It has been an enriching journey, one that has not only expanded our knowledge but also forged bonds that will endure beyond this academic pursuit.

INDEX

INTRODUCTION	6
SYSTEM REQUIREMENTS	7
PYTHON CODE	8
OUTPUT	22
BIBLIOGRAPHY	24

INTRODUCTION

SHOPIFY is an online application for purchasing groceries. In this project we tried to replicate the working of an actual online grocery shop and cover the basic functionality it. This python program is developed with simple function which enables the user of the program to create a bank account with necessary details. To sum up, the project teaches the proper use of file handling and working with multiple modes thus serving as a good reference project.

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

1. Laptop/Desktop
2. Minimum 1GB of RAM
3. Minimum 100GB of HDD

SOFTWARE REQUIREMENTS

1. Windows Operating System
2. Python 3.7 or its equivalent software

PYTHON CODE

```
'''
    This program is developed by a group of the 5 students.
    This program is a vegetable & fruit store management system.
    It allows users to create an account, sign in, and buy
    Vegetables, and view their receipts
'''
# Import necessary modules
import getpass  # Module to input passwords without echoing
import time  # Module for time-related functions
from pathlib import Path
# Initialize variables
  # Stores user purchases

# The database containing user information, vegetables, and fruits
database = {     #The Whole Database .
    'user' : {
        'aaronshenny':{
            'name' : 'Aaron Shenny',
            'password' : '123'
        },
        'user':{                          #Default user

            'name' : 'Guest',
            'password' :'root'
        },
        'aswinaravind27':{
            'name' : 'Aswin Aravind',    #User database
            'password':'aswi'


        },
        'admin' : {
            'name' : 'ADMIN',
            'password' : 'admin'
        }
    },
    'vegetables':{
        'tomato' : {
            'name' : 'Tomato',
            'price' : '₹ 48',
                'stock' : 10            #Vegetable Database
        },
        'onion': {
            'name':'Onion',
            'price':'₹ 79',
            'stock':15
        },
        'green chilli':{
            'name':'Green chilli',
```

```
            'price':'₹ 46',
            'stock':12
        },
        'beetroot':{
            'name':'Beetroot',
            'price':'₹ 34',
            'stock':14
        },
        'potato':{
            'name':'Potato',
            'price':'₹ 40',
            'stock':16
        },
        'cabbage':{
            'name':'Cabbage',
            'price':'₹ 25',
            'stock': 13
        },
        'carrot':{
            'name':'Carrot',
            'price':'₹ 39',
            'stock':17

        },
        'corn':{
            'name':'Corn',
            'price':'₹ 35',
            'stock':19
        },
        'coconut':{
            'name':'Coconut',
            'price':'₹ 37',
            'stock':16
        },
        'ginger':{
            'name':'Ginger',
            'price':'₹ 111',
            'stock':20
        },
        'elephant yam':{
            'name':'Elephant Yam',
            'price':'₹ 34',
            'stock':15
        },
        'brinjal':{
            'name':'Brinjal',
            'price':'₹ 33',
            'stock':18

        }
    },
    'fruits':{
        'apple':{
            'name':'Apple',
            'price':'₹ 190',
            'stock':21
```

```
},
'banana':{
    'name':'Banana',
    'price':'₹ 55',
    'stock': 24
},
'orange':{
    'name':'Orange',
    'price':'₹ 65',
    'stock':27
},
'mango':{
    'name':'Mango',
    'price':'₹ 89',
    'stock':13
},
'watermelon':{
    'name':'Watermelon',
    'price':'₹ 28',
    'stock':28
},
'grapes':{
    'name':'Grapes',
    'price':'₹ 150',
    'stock':12
},
'papaya':{
    'name':'Papaya',
    'price':'₹ 35',
    'stock':19
},
'guava':{
    'name':'Guava',
    'price':'₹ 89',
    'stock':11
},
'pineapple':{
    'name':'Pineapple',
    'price':'₹ 35',
    'stock':27
},
'pomegranate':{
    'name':'Pomegranate',
    'price':'₹ 189',
    'stock':30
},
'avocado':{
    'name':'Avocado',
    'price':'₹ 260',
    'stock':32
},
'dragonfruit':{
    'name':'Dragonfruit',
    'price':'₹ 299',
    'stock':31
```

```
        }
    }
}
# Function to create a new user account
def create_user(name):
    username = input('Username : ')
    if username in database['user']:  #This will check if the user
had already created account
        print('Same user has been found in our database. Please
login...')

    else:
        try:
            password = getpass.getpass(prompt = 'Create Your Account
Password : ')
        except Exception as Error:
            print('Error : ', Error)
        try:
            database['user'][username] = {
                'name': name,                              #Adds Name
and password into the database
                'password': password
            }
        except Exception as Error:
            print('Error : ', Error)
        time.sleep(2)
        print('Account created successfully...')

# Function for user sign-in
def sign_in():
    while True:
        print()
        print()
        print('\t\t\tLOGIN')
        print()
        username = input('Username : ')
        if username == 'admin':
            password1 = getpass.getpass(prompt = 'Password : ')
            if password1 == database['user'][username]['password']:
                login = True
                admin = True
                return username,login,admin
            else:
                login = False
                admin = False
                print('Incorrect Password')
                return username,login,admin

        elif username in database['user']:
#Checking given Username is matching with usernames in databse
            password1 = getpass.getpass(prompt = 'Password : ')
            if password1 == database['user'][username]['password']:
#Checking if the given password is correct with database
                time.sleep(1)
                print('Account logged in...')
                print()
```

```
                print('Welcome',database['user'][username]['name'])
                username1 = username
                login = True
#Intializing the varible as True
                admin = False
                return username,login,admin
#Returning username and login variable
                break
            else:
                login = False
#Intializing the varible as True
                admin = False
                print('Incorrect Password...')

                return username,login,admin
#Returning username and login variable
        else:
            print()
            time.sleep(1)
            print('Account not found...')
            time.sleep(1)
#If the account didnt found on the database then create_user() is
called
            print('Please sign up to continue...')
            time.sleep(1)
            print()
            print('\t\t\tSIGN-UP') #NOTE : Due to the limited
knowledge, Now creating an account will be deleted after the program
closes. Use the default username and password...')
            print()
            name  = input('Full name : ')
            create_user(name)

# Function for purchasing items
def buy(l,username,broughtitems,userbuy):
    print(broughtitems)
    if broughtitems == [] :
#brougthitems  =  list which containing the product names that user
has brougth locally
                                                        #userbuy
=  list containing both product and quantity
        brought_items = []
    else:
        #brought_items = []
        brought_items = broughtitems
    if user_buy != []:
        l = userbuy

    print()
    while True:
        print()
        item = input('Enter an item : ').lower()
#User enters the product they need
        if item == 'exit' or item == '0':
#Exiting the loop
            break
```

```
        elif item in brought_items:
            print()
#Checking the cart if the user had already brougtj
            print('Item is already in the cart!!!')
            for i in l :
                if item.title() == i[0]:
                    print(f'Product : {i[0]}')
                    print(f'Quantity : {i[1]}')
            print()
            change = input('Do you want to change the quantity ?
[yes/no] : ')
            print()                     #Asking the user if they want
to change the quantity
            if change == 'yes':
                for i in l:
                    if item.title() == i[0]:
                        if i[0].lower() in database['vegetables']:

                            product,quantity = i   #Unpacking the
tuple to change
                            quantity = float(input(f'How much kilo
of {database["vegetables"][item]["name"].lower()} do you need ? :
')) #Asking the change
                            t = product,quantity   #Packing the
tuple
                            l.remove(i)            #Removing the
existing tuple

                            l.append(t)            #Adding the new
tuple into list
                            print(f'Product :
{database["vegetables"][item]["name"]}')
                            print(f'Quantity : {quantity}')
                        elif i[0].lower() in database['fruits']:
                            product,quantity = i   #Unpacking the
tuple to change
                            quantity = float(input(f'How much kilo
of {database["fruits"][item]["name"].lower()} do you need ? : '))
#Asking the change
                            t = product,quantity   #Packing the
tuple
                            l.remove(i)            #Removing the
existing tuple

                            l.append(t)
                            print(f'Product :
{database["fruits"][item]["name"]}')
                            print(f'Quantity : {quantity}')
        elif item == '':
            print('Enter a vaild product')
        else:
            for i in l:
                if item in i[0]:
                    print()
                    print('Item is already added')
            else:
```

```
                try:
                    if item.lower() in database['vegetables'] or
item.lower() in database['fruits']:          #Checking the product
is in database
                        if item.lower() in database['vegetables'] :
                            qut = float(input(f'How much kilo of
{database["vegetables"][item]["name"].lower()} do you need ? : '))
#Asking the quantity
                            if qut < 0:
                                print('The quantity should be more
than 0')                    #Checking the quantity is more than 0
                                buy(l,username,broughtitems,userbuy)
                                break
                            if qut >
database['vegetables'][item]['stock']:                    #Checking
the given quantity is less than the stock
                                print(f'The quantity should be less
than the TOTAL STOCK, Remaining Stock :
{database["vegetables"][item]["stock"]}')
                                buy(l,username,broughtitems,userbuy)
                                break

                            brought_items.append(item)
#Adding the item into the cart
                            items =
(database['vegetables'][item]['name'],qut)
                            l.append(items)

                            database['vegetables'][item]['stock'] =
database['vegetables'][item]['stock'] - qut

                            print(f"Remaining Stocks =
{database['vegetables'][item]['stock']} kg")

                            if database['vegetables'][item]['stock']
== 0:
                                del database['vegetables'][item]


                        elif item.lower() in database['fruits']:
                            qut = float(input(f'How much kilo of
{database["fruits"][item]["name"].lower()} do you need ? : '))
                            if qut < 0:
                                print('The quantity should be more
than 0')                    #Checking the quantity is more than 0
                                buy(l,username,broughtitems,userbuy)
                                break
                            if qut >
database['fruits'][item]['stock']:                    #Checking the
given quantity is less than the stock
                                print(f'The quantity should be less
than the TOTAL STOCK, Remaining Stock :
{database["fruits"][item]["stock"]}')
                                buy(l,username,broughtitems,userbuy)
                                break
```

```
                            brought_items.append(item)
#Adding the item into the cart
                            items =
(database['fruits'][item]['name'],qut)
                            l.append(items)

                            database['fruits'][item]['stock'] =
database['fruits'][item]['stock'] - qut

                            print(f"Remaining Stocks =
{database['fruits'][item]['stock']} kg")

                            if database['fruits'][item]['stock'] ==
0:
                                del database['fruits'][item]
                    else:
                        print('Item not found')

                except ValueError:
#Exception handling
                    print('Please enter a valid value...')

    if username in user_buy:
        existing_items = user_buy[username]
        l1 = existing_items + l
        user_buy[username] = l1
        addInfo(user_buy)

        return user_buy, l , brought_items

    else:
        user_buy[username] = l
        addInfo(user_buy)

        return user_buy,l , brought_items



#Function for listing the items
def list1(database):
    vegetable_data = database.get('vegetables')
    fruits_data = database.get('fruits')

    if not vegetable_data:
        print("No vegetable data found!")
#Checking if the database is empty or not
        return
    if not fruits_data:
        print("No vegetable data found!")
#Checking if the database is empty or not
        return
    print()
    print("-----------------------------------------\t\t ----------
-------------------------------")
    print("|   Vegetable   |     Price     | Stock  |\t\t|    Fruits
|     Price     | Stock  |")
```

```
    print("-----------------------------------------\t\t ----------
------------------------------")

    veg_keys = list(database['vegetables'].keys())

    fru_keys = list(database['fruits'].keys())

    for i, j in zip(veg_keys,fru_keys):
        veg_name = database['vegetables'][i]['name'].ljust(15)
        veg_price = database['vegetables'][i]['price'].ljust(15)
        veg_stock = str(database['vegetables'][i]['stock']).ljust(8)

        fruit_name = database['fruits'][j]['name'].ljust(15)
        fruit_price = database['fruits'][j]['price'].ljust(15)
        fruit_stock = str(database['fruits'][j]['stock']).ljust(8)


print(f'|{veg_name}|{veg_price}|{veg_stock}|\t\t|{fruit_name}|{fruit
_price}|{fruit_stock}|')
    print("-----------------------------------------\t\t ----------
------------------------------")

def receipt(username,brought_items,broughtitems,userbuy):
#Function for printing the receipt
    confirm =  input('Anything else ? : ').lower()
#Asking the user if they want to buy anything else
    if confirm == 'yes':
        l =  userbuy
        userbuy, brought_items ,broughtitems =
buy(userbuy,username,broughtitems,brought_items)
    total_amount = 0  # Initialize the total amount variable

    print()
    print('=' * 70)
    print('RECEIPT'.center(70))
    print('=' * 70)
    time2 = time.asctime()
#Getting the current time

    print('Name :
',database['user'][username]['name'],'\t\t\t','Date : ',time2)
    print('=' * 70)


print(''.ljust(8),'ITEM'.ljust(19),'RATE'.ljust(14),'QUANTITY'.ljust
(17),'TOTAL'.ljust(8))
    print('=' * 70)

    for i in  brought_items:
        product_name, quantity = i
        price_per_kilo = 0

        # Check if the product is a vegetable or a fruit
        if product_name.lower() in database['vegetables']:
            price_per_kilo =
float(database['vegetables'][product_name.lower()]['price'][2:])  #
```

```
Extract price per kilo
        elif product_name.lower() in database['fruits']:
            price_per_kilo =
float(database['fruits'][product_name.lower()]['price'][2:])  #
Extract price per kilo

        total_price = price_per_kilo * quantity
        total_amount += total_price

        print(product_name.ljust(20)
,'|'.ljust(3),'₹',str(price_per_kilo).ljust(5), "/kg".ljust(8)
,'|'.ljust(4),str(quantity).ljust(3) ,"kg".ljust(6)
,'|'.ljust(3),'₹',str(total_price).ljust(5))

    print()
    print('=' * 70)
    print('Total Amount :','₹',total_amount)

def login_checker(login):
    if login != True:
        main()

def adminf():
    print()
    print('~~~~~~~~~~~')
    print('ADMIN PANEL')
    print('~~~~~~~~~~~')
    print()
    print('1. Change the rate of the product')
    print('2. Change the stock of the product')
    print('3. ORDERS')
    print('0. Exit admin panel')
    while True:
        print()
        try:
            choice = int(input('Enter the choice : '))
            if choice == 1:
                prodName = input('Product Name : ').lower()
                if prodName in database['vegetables'] or prodName in
database['fruits']:
                    if prodName in database['vegetables']:
                        for i in database['vegetables']:
                            if i == prodName:
                                rate =  input('Enter the revised
rate : ')

database['vegetables'][prodName]['price'] = '₹ '+rate
                                print('Rate updated
successfully...')
                                print(f'PRODUCT :
{database["vegetables"][prodName]["name"]}')
                                print(f'RATE :
{database["vegetables"][prodName]["price"]}')
                    elif prodName in database['fruits']:
                        for i in database['fruits']:
                            if i == prodName:
```

```
                                rate =  input('Enter the revised
rate : ')

database['fruits'][prodName]['price'] = '₹ '+rate
                                print('Rate updated
successfully...')
                                print(f'PRODUCT :
{database["fruits"][prodName]["name"]}')
                                print(f'RATE :
{database["fruits"][prodName]["price"]}')
                    else:
                        print('404 Item Not Found')
                else:
                    print('404 Item Not Found')
            elif choice == 2:
                prodName = input('Product Name : ').lower()
                if prodName in database['vegetables'] or prodName in
database['fruits']:
                    if prodName in database['vegetables']:
                        for i in database['vegetables']:
                            if i == prodName:
                                stock =  input('Enter the revised
stock number : ')

database['vegetables'][prodName]['stock'] = stock
                                print('Stock updated
successfully...')
                                print(f'PRODUCT :
{database["vegetables"][prodName]["name"]}')
                                print(f'STOCK :
{database["vegetables"][prodName]["stock"]}')
                    elif prodName in database['fruits']:
                        for i in database['fruits']:
                            if i == prodName:
                                stock =  input('Enter the revised
stock number : ')

database['fruits'][prodName]['stock'] = stock
                                print('Stock updated
successfully...')
                                print(f'PRODUCT :
{database["fruits"][prodName]["name"]}')
                                print(f'STOCK :
{database["fruits"][prodName]["stock"]}')
                    else:
                        print('404 Item Not Found')

                else:
                    print('404 Item Not Found')
            elif choice ==3:
                print()
                print('ORDERS')
                if not getInfo('user_buy'):
                    print('No recent Orders')
                else:
                    user_buy1 = eval(getInfo('user_buy'))
```

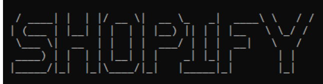
```
                    #print(user_buy1)
                    for i in user_buy1:
                        print()
                        #print(i)
                        print('|-----------------------------------
|')
                        print('|'.ljust(10),'USERNAME :
',i.upper().ljust(13),'|')
                        print('|-----------------------------------
|')

print('|'.ljust(8),'ITEM'.ljust(15),'QUANTITY'.ljust(11),'|')
                        print('|-----------------------------------
|')


                        for j in user_buy1[i]:
                            #print(' ',j[0].ljust(),j[1])
                            print('|',j[0].ljust(17)
,'|'.ljust(8),'₹',str(j[1]).ljust(5),'|')
                        print('|-----------------------------------
|')
            elif choice == 0 :
                break
            else:
                print('Invalid Choice')
        except ValueError as Error:
            print('Enter the valid input')

def addInfo(var):
    for name, value in globals().items():  # Use locals() for local
variables
        if value is var:

            var_name = name
    f = open(Path('data.txt'),'w')
    f.write(f'{var_name} = {var}\n')
    f.close()
def getInfo(var):
    file_path = Path('data.txt')
    for name, value in globals().items():  # Use locals() for local
variables
        if value is var:

            var_name = name
    if not file_path.exists():
        var_name = {}
        return  var_name # or handle as needed if the file doesn't
exist
    with open(Path('data.txt'), 'r') as file:
    # Read each line in the file
        for line in file:
            # Check if the line contains the variable you want
            if line.startswith(var):
                # Split the line at '=' to get the value part
                variable_value = line.split('=')[-1].strip()
```

```
                #variable_value = eval(variable_value)
                return variable_value



print()
print('='*55)
print()
print('  / ____| |  | |/ __ \|  __ \_   _|  ____\ \   / /')
print(' | (___ | |__| | |  | | |__) || | | |__   \ \_/ / ')
print('  \___ \|  __  | |  | |  ___/ | | |  __|   \   /  ')
print('  ____) | |  | | |__| | |    _| |_| |       | |   ')
print(' |_____/|_|  |_|\____/|_|   |_____|_|       |_|   ')
print()
print('='*55)
if not getInfo('user_buy'):
    user_buy={}
   # print(user_buy)
else:
    user_buy =  eval(getInfo('user_buy'))
print()

time.sleep(1)
n=0
def main():

    username = None
    while True:

            time.sleep(1)
            username,login, admin = sign_in()

            if login == False:
                login_checker(login)
            else:

                time.sleep(1)
                if admin == False :

                    list1(database)
                    print()
                    buy_accept =  input('Wanna buy something from
our store ??? [yes/no] : ').lower()   #Asking the user if they want
to buy anything..reconfirming
                    if buy_accept == 'yes':
                        time.sleep(1)
                        l = []
                        print()
                        print('NOTE : Please enter "0" or "exit"
once you have completed adding the products.')
                        broughtitems = []
                        userbuy = []
                        userbuy, l,broughtitems =
buy(l,username,broughtitems,userbuy)
```

```
                        if user_buy[username] == []:
                            pass
                        else:

                            receipt(username,l,broughtitems,userbuy)
                            break
                    else:
                        time.sleep(1)
                        print()
                        print('\t\t\tThank you for coming!!!')
                        time.sleep(5)
                    break
                elif admin == True:
                    adminf()
                    break
                else:
                    print('ERROR')


if __name__ == "__main__":
    main()
    while True:
        time.sleep(2)
        print()
        choice = input("Enter 'q' to quit or any other key to
proceed to the next customer : ")  #Asking the user if they want to
quit or proceed to the next customer
        print()
        if choice.lower() == 'q':
            print('\t\t\tThank you for coming!!!')
            print('\t\t\t   Visit again!!!')
            print()
            print("Exiting the program...")
            print()
            break
        else:
            print('NEXT CUSTOMER PLEASE...')
            time.sleep(2)
            main()

##END OF THE PROGRAM!!
```

OUTPUT

```
=======================================================
```



```
=======================================================


                        LOGIN

Username : bivia

Account not found...
Please sign up to continue...

                        SIGN-UP

Full name : Bivia
Username : bivia
Create Your Account Password :
Account created successfully...


                        LOGIN

Username : bivia
Password :
Account logged in...

Welcome Bivia

------------------------------------------              ------------------------------------------
|   Vegetable   |      Price      | Stock  |            |     Fruits     |      Price      | Stock  |
------------------------------------------              ------------------------------------------
|Tomato         |₹ 48             |10      |            |Apple           |₹ 190            |21      |
|Onion          |₹ 79             |15      |            |Banana          |₹ 55             |24      |
|Green chilli   |₹ 46             |12      |            |Orange          |₹ 65             |27      |
|Beetroot       |₹ 34             |14      |            |Mango           |₹ 89             |13      |
|Potato         |₹ 40             |16      |            |Watermelon      |₹ 28             |28      |
|Cabbage        |₹ 25             |13      |            |Grapes          |₹ 150            |12      |
|Carrot         |₹ 39             |17      |            |Papaya          |₹ 35             |19      |
|Corn           |₹ 35             |19      |            |Guava           |₹ 89             |11      |
|Coconut        |₹ 37             |16      |            |Pineapple       |₹ 35             |27      |
|Ginger         |₹ 111            |20      |            |Pomegranate     |₹ 189            |30      |
|Elephant Yam   |₹ 34             |15      |            |Avocado         |₹ 260            |32      |
|Brinjal        |₹ 33             |18      |            |Dragonfruit     |₹ 299            |31      |
------------------------------------------              ------------------------------------------

Wanna buy something from our store ??? [yes/no] : yes

NOTE : Please enter "0" or "exit" once you have completed adding the products.
[]

Enter an item : tomato
How much kilo of tomato do you need ? : 4
Remaining Stocks = 6.0 kg

Enter an item : apple
How much kilo of apple do you need ? : 5
Remaining Stocks = 16.0 kg

Enter an item : apple

Item is already in the cart!!!
Product : Apple
Quantity : 5.0

Do you want to change the quantity ? [yes/no] : yes

How much kilo of apple do you need ? : 10
Product : Apple
Quantity : 10.0

Enter an item : ginger
How much kilo of ginger do you need ? : 25
The quantity should be less than the TOTAL STOCK, Remaining Stock : 20
[]


Enter an item : 0
Anything else ? : no

======================================================================
                               RECEIPT
======================================================================
Name :  Bivia                    Date :  Sun Jan 21 17:33:57 2024
======================================================================
         ITEM                RATE             QUANTITY          TOTAL
======================================================================
Tomato                |   ₹ 48.0  /kg      |    4.0 kg      |   ₹ 192.0
Apple                 |   ₹ 190.0 /kg      |    10.0 kg     |   ₹ 1900.0

======================================================================
Total Amount : ₹ 2092.0

Enter 'q' to quit or any other key to proceed to the next customer : q

                        Thank you for coming!!!
                           Visit again!!!
```

```
=====================================================

  / ____| |  | |/ __ \|  __ \_   _| ____\ \   / /
 | (___ | |__| | |  | | |__) || | | |__   \ \_/ /
  \___ \|  __  | |  | |  ___/ | | |  __|   \   /
  ____) | |  | | |__| | |    _| |_| |       | |
 |_____/|_|  |_|\____/|_|   |_____|_|       |_|

=====================================================



                      LOGIN

Username : admin
Password :

~~~~~~~~~~~~
ADMIN PANEL
~~~~~~~~~~~~

1. Change the rate of the product
2. Change the stock of the product
3. ORDERS
0. Exit admin panel

Enter the choice : 1
Product Name : ginger
Enter the revised rate : 20
Rate updated successfully...
PRODUCT : Ginger
RATE : ₹ 20

Enter the choice : 2
Product Name : tomato
Enter the revised stock number : 300
Stock updated successfully...
PRODUCT : Tomato
STOCK : 300

Enter the choice : 3

ORDERS

|-----------------------------------|
|          USERNAME :  USER         |
|-----------------------------------|
| Tomato            |       ₹ 3.0   |
| Apple             |       ₹ 3.0   |
| Apple             |       ₹ 5.0   |
|-----------------------------------|

|-----------------------------------|
|          USERNAME :  BIVIA        |
|-----------------------------------|
|        ITEM           QUANTITY    |
|-----------------------------------|
| Tomato            |       ₹ 4.0   |
| Apple             |       ₹ 10.0  |
| Tomato            |       ₹ 4.0   |
| Apple             |       ₹ 10.0  |
|-----------------------------------|

Enter the choice : 0

Enter 'q' to quit or any other key to proceed to the next customer : q

                     Thank you for coming!!!
                        Visit again!!!

Exiting the program...
```

BIBLIOGRAPHY

1) https://stackoverflow.com/questions/9632995/how-to-easily-print-ascii-art-text
2) https://stackoverflow.com/questions/41816268/printing-multiple-dictionary-keys- on-one-line
3) https://stackoverflow.com/questions/53513/how-do-i-check-if-a-list-is-empty&usg=AOvVaw3N6uqwooe6LXI7GipBGFOY
4) https://stackoverflow.com/questions/17610732/error-dictionary-update-sequence- element-0-has-length-1-2-is-required-on-dj
5) https://www.geeksforgeeks.org/how-to-open-and-close-a-file-in-python/
6) https://www.geeksforgeeks.org/python-check-if-list-empty-not/
7) https://www.geeksforgeeks.org/python-nested-dictionary/
8) https://docs.python.org/3/library/time.html
9) https://docs.python.org/3/library/getpass.html
10) https://www.geeksforgeeks.org/python-ways-to-remove-a-key-from-dictionary/