DELHI PUBLIC SCHOOL, DWARKA NEW DELHI

ACADEMIC SESSION: 2023-24

PROJECT REPORT ON
GROCERY MANAGEMENT SYSTEM

FOR **COMPUTER SCIENCE**

AISCEE 2023-24

NAME: ARUN KUMAR

ROLL NUMBER: 21309021

CLASS: XII

SUBJECT: COMPUTER SCIENCE

SUB. CODE: 083

PROJECT GUIDE: <Name of Your computer Teacher>

DELHI PUBLIC SCHOOL, DWARKA NEW DELHI

CERTIFICATE

This is to certify that cadet **ARUN KUMAR** ,Roll no: 21309021

has successfully completed the project work entitled **GROCERY STORE MANAGEMENT SYSTEM** in the subject Computer science (083) laid down in the regulations of CBSE for the purposes of Practical examination in **Class XII** to be held in **Delhi Public School, Dwarka, New Delhi.**

(Name of your computer teacher)

PGT Computer Science

EXAMINER:

Name:________

Signature:

TABLE OF CONTENTS(TOC)

ACKNOWLEDGEMENT

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I express my deep sense of gratitude to the luminary The Principal who has been continuously motivating and extending their helping hand to us.

I express my sincere thanks to the academician, the Vice Principal, for constant encouragement and the guidance provided during this project.

I am overwhelmed to express my thanks to The Administrative Officer for providing me with infrastructure and moral support while carrying out this project in the school.

My sincere thanks to Mr. <Name of your computer teacher>, Master In-charge, A guide, Mentor all the above a friend, who critically reviewed my project and helped in solving each and every problem, occurred during implementation of the project

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

PROJECT ON GROCERY STORE MANAGEMENT SYSTEM

INTRODUCTION

This project is all about software for a grocery store. It helps the Store Owner to have a full-fledged control over his/her store. It adds a new stock, updates an existing stock and helps consumers view and place orders.

OBJECTIVES OF THE PROJECT

The objective of this project is to let the students apply the programming knowledge into a real-world situation/problem and expose the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.

2. Apply object-oriented programming principles effectively when developing small to medium sized projects.

3. Write effective procedural code to solve small to medium sized problems.

4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.

5. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

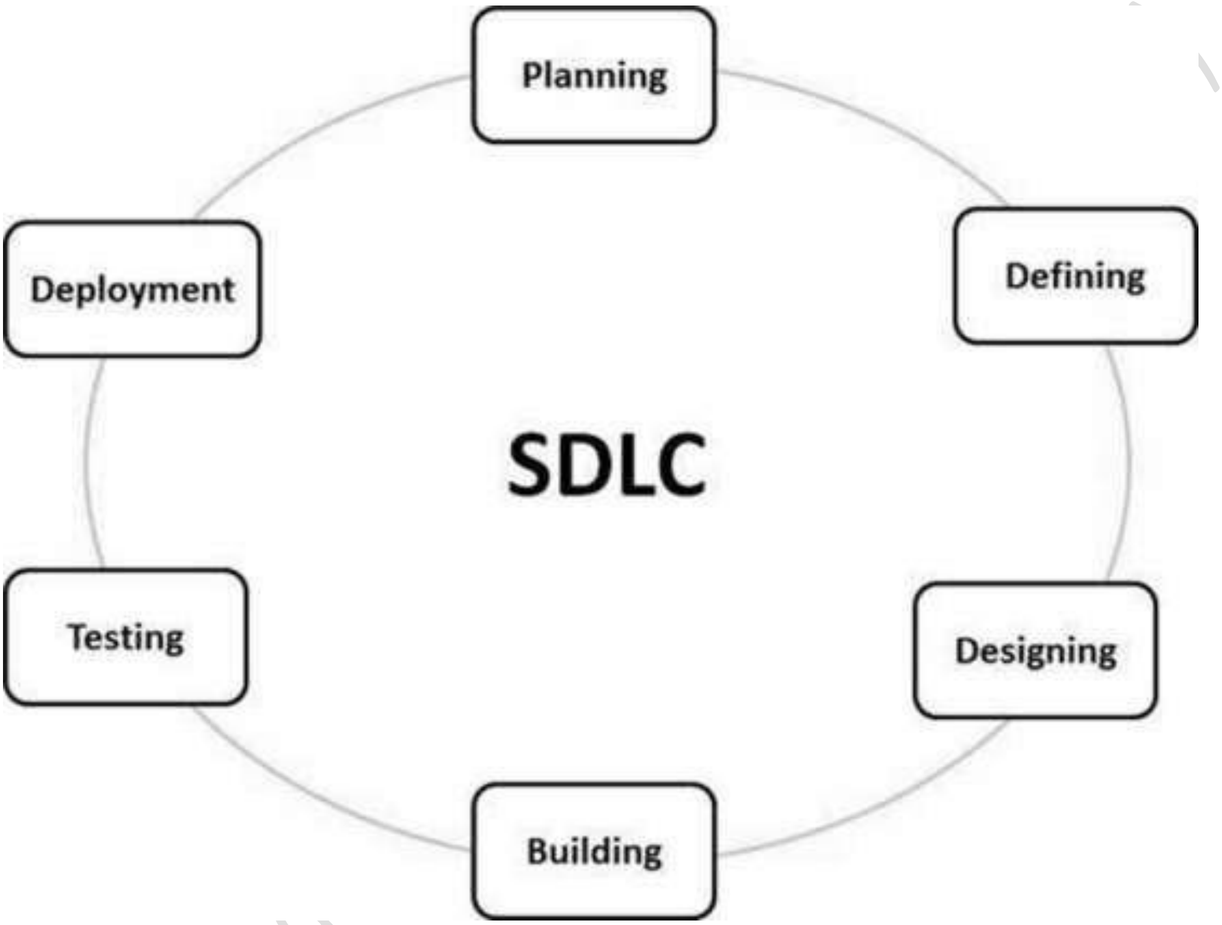
PROPOSED SYSTEM

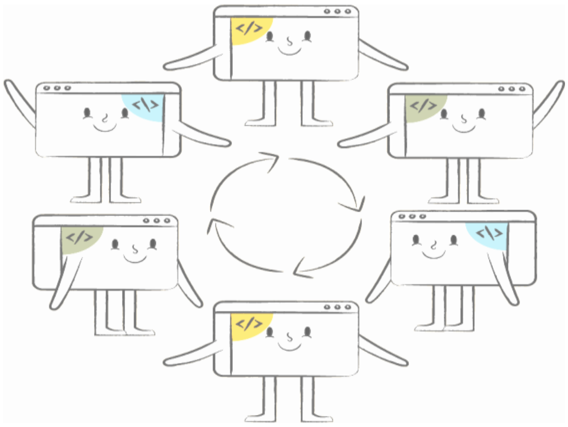
Today one cannot afford to rely on the fallible human beings who really want to stand against today's merciless competition where the not too wise saying "to err is human '' is no longer valid, it's outdated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a much-sophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomization various organizations. Many software products working is now in markets, which have helped in making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paperwork had to be done but now software products in this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done.

This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it's an age of computers and automating such an organization gives a better look.

SYSTEM DEVELOPMENT LIFE CYCLE(SDLC)





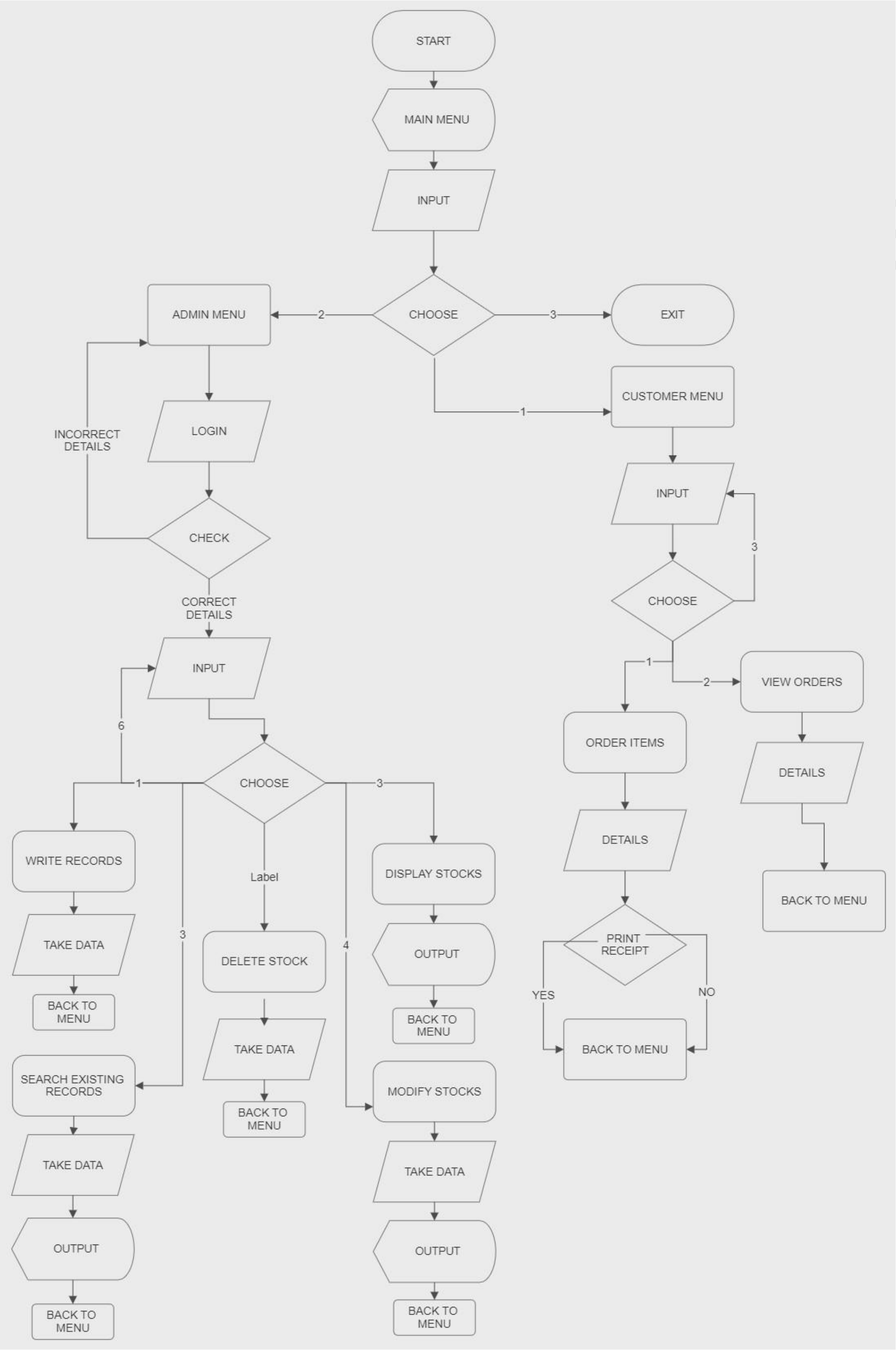
MODULES USED AND THEIR PURPOSE

Pickle module:The pickle module implements binary protocols for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as “serialization”, “marshalling,” 1 or “flattening”; however, to avoid confusion, the terms used here are “pickling” and “unpickling”.

Time module:time.sleep()-Suspend execution of the calling thread for the given number of seconds. The argument may be a floating point number to indicate a more precise sleep time.If the sleep is interrupted by a signal and no exception is raised by the signal handler, the sleep is restarted with a recomputed timeout.The suspension time may be longer than requested by an arbitrary amount, because of the scheduling of other activity in the system.

OS module:os.startfile()-When operation is not specified, this acts like double-clicking the file in Windows Explorer, or giving the file name as an argument to the start command from the interactive command shell: the file is opened with whatever application (if any) its extension is associated.When another operation is given, it must be a “command verb” that specifies what should be done with the file. Common verbs documented by Microsoft are 'open', 'print' and 'edit' (to be used on files) as well as 'explore' and 'find' (to be used on directories).When launching an application, specify arguments to be passed as a single string. This argument may have no effect when using this function to launch a document.

FLOW CHART


START
MAIN MENU
INPUT
CHOOSE
ADMIN MENU
2
3
EXIT
1
CUSTOMER MENU
LOGIN
INCORRECT DETAILS
CHECK
CORRECT DETAILS
INPUT
INPUT
3
CHOOSE
1
2
VIEW ORDERS
6
ORDER ITEMS
DETAILS
CHOOSE
1
3
WRITE RECORDS
Label
DISPLAY STOCKS
DETAILS
BACK TO MENU
TAKE DATA
3
4
OUTPUT
PRINT RECEIPT
DELETE STOCK
BACK TO MENU
YES
NO
BACK TO MENU
BACK TO MENU
TAKE DATA
SEARCH EXISTING RECORDS
BACK TO MENU
MODIFY STOCKS
TAKE DATA
TAKE DATA
OUTPUT
OUTPUT
BACK TO MENU
BACK TO MENU

SOURCE CODE

```
import pickle

import time

import os


def intro():

    print("="*80)

    print("{: ^80s}".format("GROCERY STORE"))

    print("{: ^80s}".format("MANAGEMENT SYSTEM"))

    print("{: ^80s}".format("PROJECT"))

    print("="*80)

    print()


def main_menu():

    time.sleep(1)

    print("MAIN MENU")

    print("1. CUSTOMER MENU")

    print("2. ADMIN MENU")

    print("3. EXIT")


def customer_menu():

    time.sleep(1)

    print("CUSTOMER MENU")

    print("1. ORDER ITEMS")

    print("2. VIEW ORDERS")

    print("3. BACK TO MAIN MENU")


def order_items():
```

```
time.sleep(1)
fh=open("aorders.dat","ab")
nme=input("Please enter your name:")
lst=[]
while True:
     itm_name=input("Enter the name of item:")
     stck=open("stocks.dat","rb")
     while True:
          try:
              s=pickle.load(stck)
              if s[1]==itm_name:
                  itm_qty=int(input("Enter the quantity:"))
                  if itm_qty<=s[3]:
                      a=[itm_name,s[2],itm_qty]
                      lst.append(a)
                  else:
                      print('Stock is low.')

          except EOFError:
               stck.close()
               break
     chk=input("Do you want to add more?(Y/N):")
     if chk=="Y":
         continue
     else:
         break

print("Thanks for placing the order")
lst.append(nme)
ans=input("Do you want a receipt?Y/N:")
```

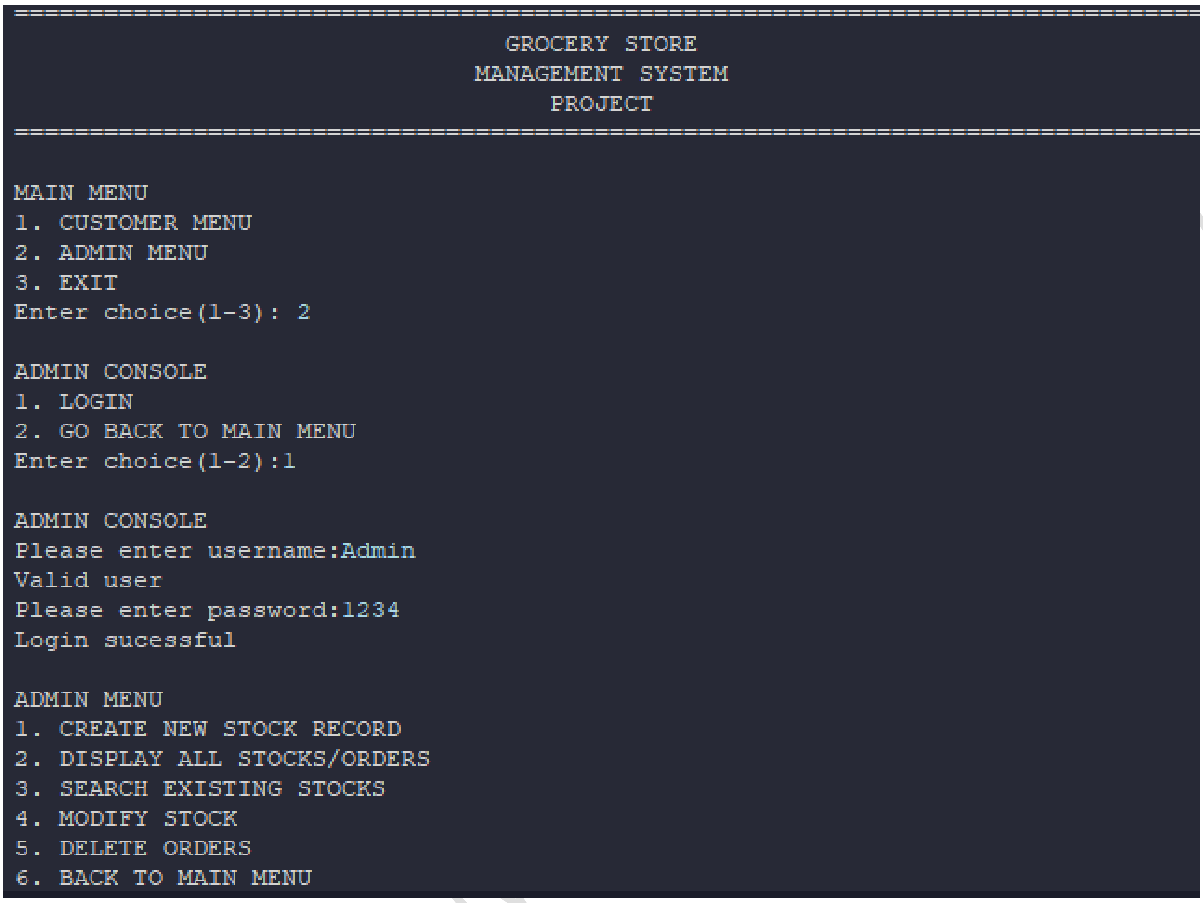
```
if ans=="Y":
    g=open("order.txt","w")
    g.write("\n GROCERY STORE \n")
    g.write("\n user:admin \n")
    from datetime import datetime
    now = datetime.now()
    dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
    g.write("\n Date and time:")
    g.write(dt_string )
    total=0
    ln=len(lst)
    for j in range(0,ln-1):
        q = lst[j][1]
        r = lst[j][2]
        ttl = q*r
        total=total +ttl
        ab=lst[j]
        abc=str(ab)
        g.write(abc)
    acd=str(total)
    g.write("\n TOTAL=")
    g.write(acd)
    g.write("\n THANKS FOR SHOPPING WITH US \n")
    os.startfile("order.txt", "print")
    g.close()

if ans=="N":
    print("YOUR ORDER HAS BEEN PLACED,THANKS FOR SHOPPING")
pickle.dump(lst,fh)
```

```
    fh.close()

def view_orders():
    time.sleep(1)
    fh=open("aorders.dat","rb")
    nme=input("Please enter your name:")
    vl='t'
    while True:
        try:
            f=pickle.load(fh)
            if f[-1]==nme:
                print(f)
                vl='f'

        except EOFError:
            break
    if vl=='t':
        print("No record found")
    fh.close()

def admn_check():
    time.sleep(1)
    print("\nADMIN CONSOLE")
    while True:
        chk=input("Please enter username:")
        if chk=="Admin":
            print("Valid user")
            pas=input("Please enter password:")
            if pas=="1234":
                print("Login successful")
                admn_menu()
                break
            else:
                print("Please enter correct details")
                pass
        else:
            print("Please enter correct details")
            pass

def admin_menu():
    time.sleep(1)
    print("\nADMIN MENU")
    print("1. CREATE NEW STOCK RECORD")
    print("2. DISPLAY ALL STOCKS/ORDERS")
    print("3. SEARCH EXISTING STOCKS")
    print("4. MODIFY STOCK")
    print("5. DELETE ORDERS")
    print("6. BACK TO MAIN MENU")

def write_record():
    stck=open("stocks.dat","ab")
    while True:
        srno=int(input("Enter serial number:"))
        itmname=input("Enter name of item:")
        itmprice=int(input("Enter the price:"))
        qty=int(input("Enter the quantity:"))
        b=[srno,itmname,itmprice,qty]
        pickle.dump(b,stck)
        chk=input("Do you want to enter more records?Y/N:")
        if chk=="Y":
            continue
        else:
            stck.close()
            break
    time.sleep(4)

def read_records():
    chk=input("What do you want to view?(stocks/orders):")
    if chk=="stocks":
        stck=open("stocks.dat","rb")
        print("[SrNo.,Item_name,Price,Quantity]")
        while True:
            try:
                s=pickle.load(stck)
                print(s)
            except EOFError:
                stck.close()
                break
        time.sleep(4)
    if chk=="orders":
        fh=open("aorders.dat","rb")
        while True:
            try:
                f=pickle.load(fh)
                print(f)
            except EOFError:
                break
        fh.close()
        time.sleep(4)
```

```
def search_record():

    fh=open("stocks.dat","rb")

    ab=open("aorders.dat","rb")

    val="t"

    search=input("Enter item name/serial number/customer name to
search?:")

    for i in search:

        if i in "1234567890":

            al=int(search)

        else:

            al=search


    while True:

        try:

            f=pickle.load(fh)

            if f[1]==search or f[0]==al:

                print(f)

                val="f"

                time.sleep(3)

        except EOFError:

            if val=="t":

                print("Item not found in stocks")

                val="d"

            break

    fh.close()

    if val=="d":

        while True:

            try:

                a=pickle.load(ab)

                if a[-1]==search:
                    val=="f"
                    print(a)
                    time.sleep(3)
            except EOFError:
                if val=="d":
                    print("Item not found in orders")
                    val="c"
                break
    ab.close()
    if val=="c":
        print("Item not present")
        time.sleep(3)

def modify_record():
    time.sleep(1)
    while True:
        fh=open("stocks.dat","rb+")
        sr_no=int(input("Enter serial number change:"))
        flag=0
        while True:
            try:
                f=pickle.load(fh)
                if f[0]==sr_no:
                    f[1]=input("Enter new name:")
                    f[2]=input("Enter new price:")
                    f[3]=input("Update quantity:")
                    flag=1
                    print("Data updated successfully")
                    time.sleep(2)
                sd=[sr_no,f[1],f[3],f[3]]
                pickle.dump(sd,fh)
                print(sd)
            except EOFError:
                break
        if flag==0:
            print("NO value found")
        ask=input("Do you want to update more values?(Y/N):")
        if ask=="Y":
            continue
        else:
            fh.close()
            break


def admn_menu():
    time.sleep(1)
    while True:
            admin_menu()
            echoice = input('Enter choice(1-6):')
            print()
            if echoice == '1':
                write_record()
            elif echoice == '2':
                read_records()
            elif echoice == '3':
                search_record()
            elif echoice == '4':
                modify_record()
            elif echoice == '5':
                modify_record()
            elif echoice == '6':
                break
            else:
                print('Invalid input !!!\n')


def main():
    intro()
    while(True):
        main_menu()
        choice = input('Enter choice(1-3): ')
        print()

        if choice == '1':
            while True:
                customer_menu()
                rchoice = input('Enter choice(1-3): ')
                print()
                if rchoice == '1':
                    order_items()
                elif rchoice == '2':
                    view_orders()
                elif rchoice == '3':
                    break
                else:
                    print('Invalid input !!!\n')
                print()

        elif choice == '2':
            print("ADMIN CONSOLE")
            print("1. LOGIN")
            print("2. GO BACK TO MAIN MENU")
            hchoice=input("Enter choice(1-2):")
            if hchoice=="1":
                admn_check()
            elif hchoice=="2":
                break
            else:
                print("Invalid input !!!\n")

        elif choice == '3':
            print('Thanks for using the Grocery Management System')
            break
        else:
            print('Invalid input!!!')
            print()


main()
```

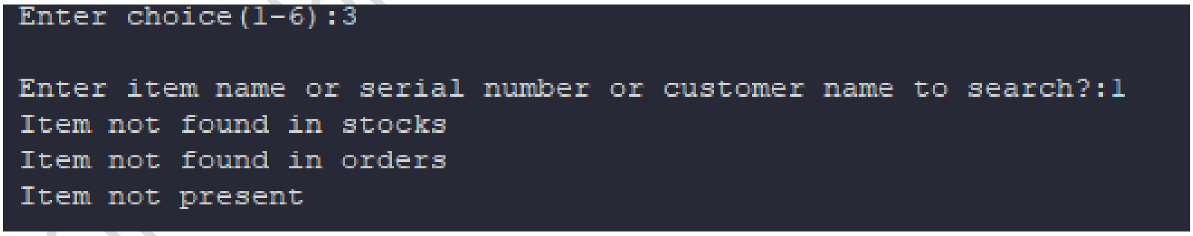
OUTPUT

Menu screen:

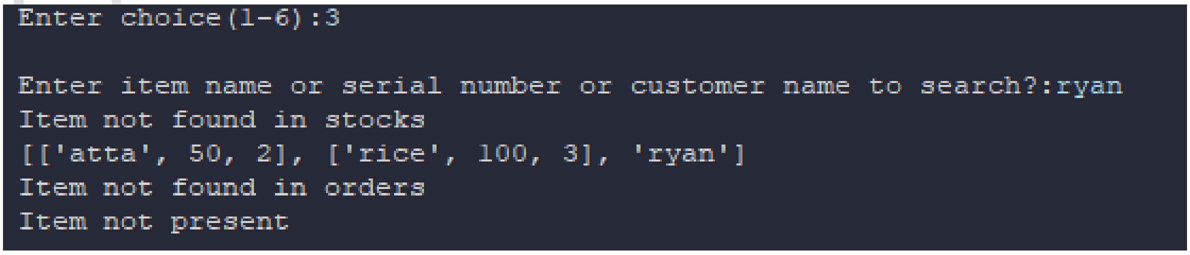


```
================================================================================
                              GROCERY STORE
                            MANAGEMENT SYSTEM
                                 PROJECT
================================================================================

MAIN MENU
1. CUSTOMER MENU
2. ADMIN MENU
3. EXIT
Enter choice(1-3): 2

ADMIN CONSOLE
1. LOGIN
2. GO BACK TO MAIN MENU
Enter choice(1-2):1

ADMIN CONSOLE
Please enter username:Admin
Valid user
Please enter password:1234
Login sucessful

ADMIN MENU
1. CREATE NEW STOCK RECORD
2. DISPLAY ALL STOCKS/ORDERS
3. SEARCH EXISTING STOCKS
4. MODIFY STOCK
5. DELETE ORDERS
6. BACK TO MAIN MENU
```

Searching existing stock:

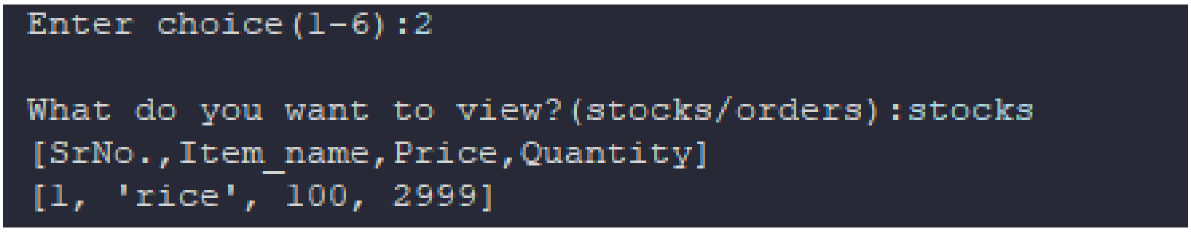


```
Enter choice(1-6):3

Enter item name or serial number or customer name to search?:1
Item not found in stocks
Item not found in orders
Item not present
```



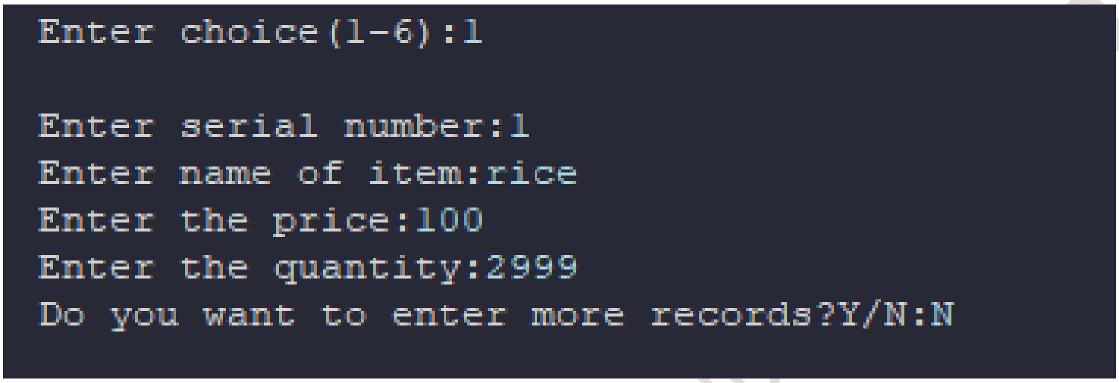
```
Enter choice(1-6):3

Enter item name or serial number or customer name to search?:ryan
Item not found in stocks
[['atta', 50, 2], ['rice', 100, 3], 'ryan']
Item not found in orders
Item not present
```

Displaying all available stocks:



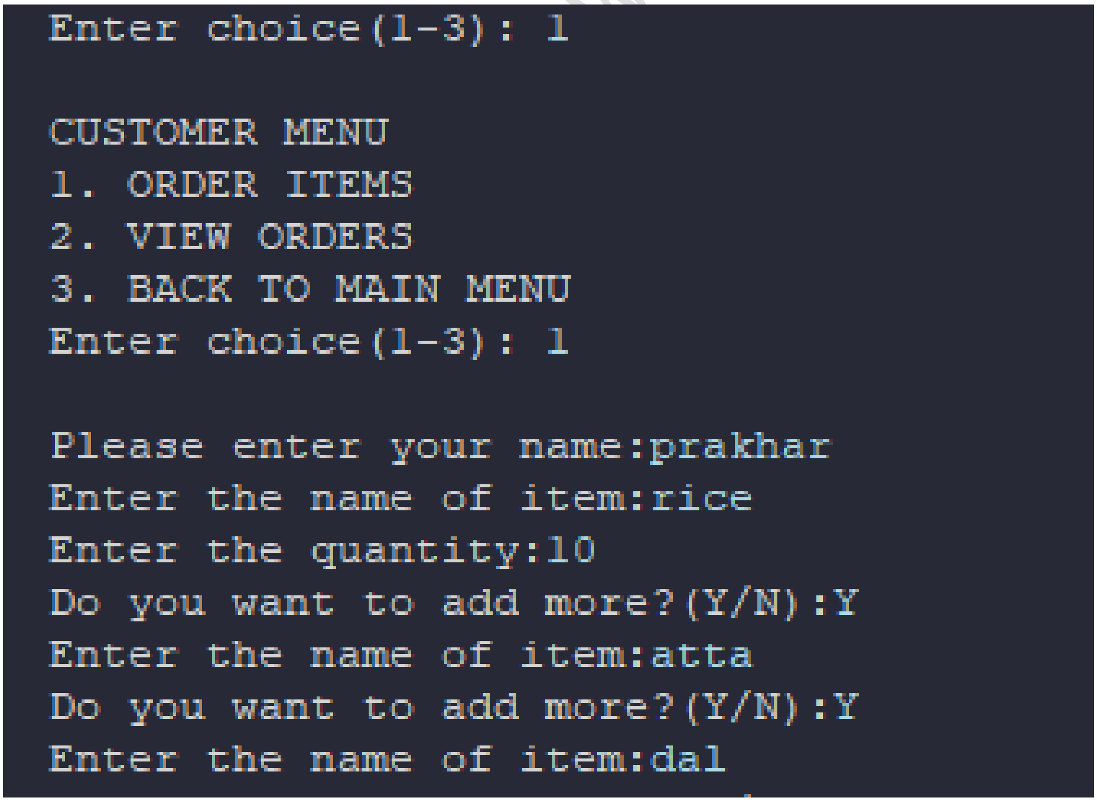
```
Enter choice(1-6):2

What do you want to view?(stocks/orders):stocks
[SrNo.,Item_name,Price,Quantity]
[1, 'rice', 100, 2999]
```

Adding data via admin console:



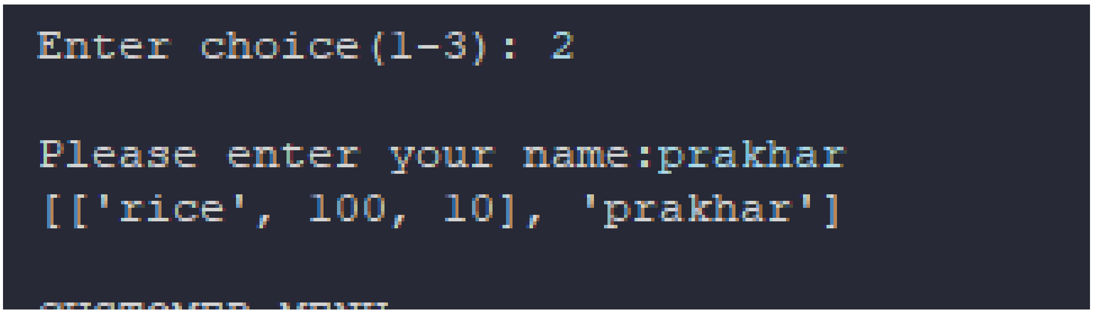
```
Enter choice(1-6):1

Enter serial number:1
Enter name of item:rice
Enter the price:100
Enter the quantity:2999
Do you want to enter more records?Y/N:N
```

Placing order:



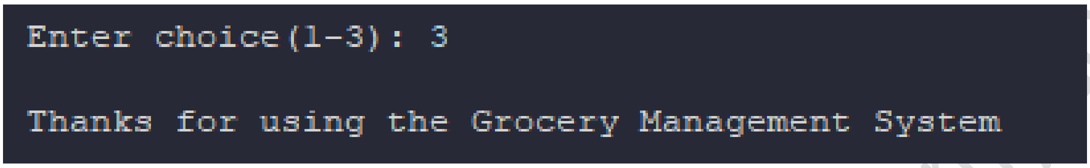
```
Enter choice(1-3): 1

CUSTOMER MENU
1. ORDER ITEMS
2. VIEW ORDERS
3. BACK TO MAIN MENU
Enter choice(1-3): 1

Please enter your name:prakhar
Enter the name of item:rice
Enter the quantity:10
Do you want to add more?(Y/N):Y
Enter the name of item:atta
Do you want to add more?(Y/N):Y
Enter the name of item:dal
```

Viewing orders:



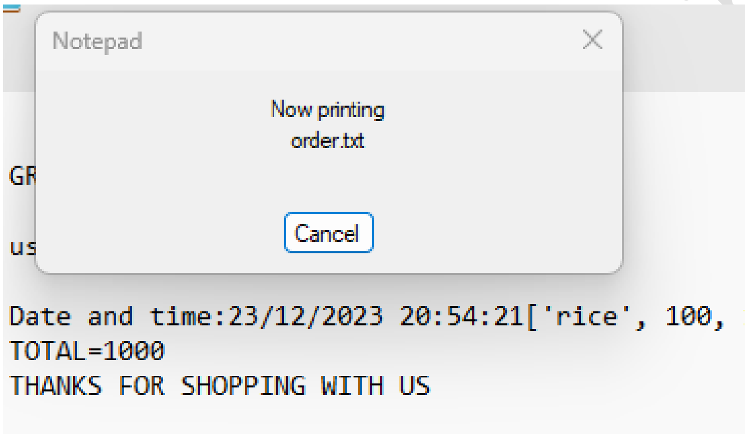
```
Enter choice(1-3): 2

Please enter your name:prakhar
[['rice', 100, 10], 'prakhar']

CUSTOMER MENU
```

Simple and easy exiting sequence:

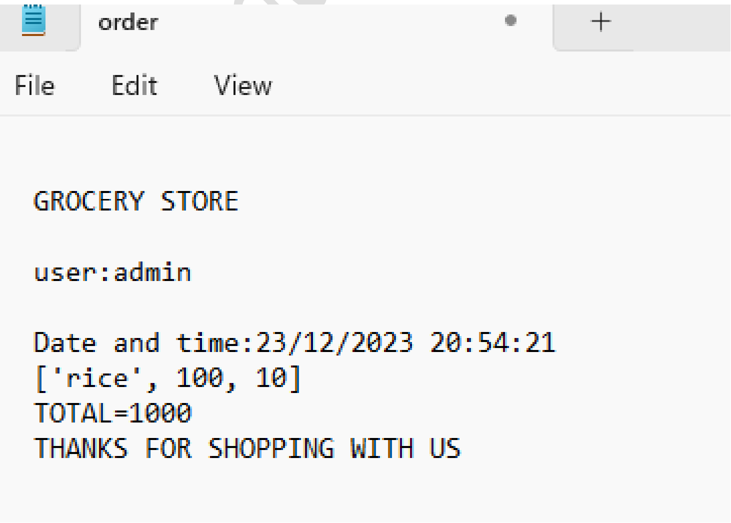


```
Enter choice(1-3): 3

Thanks for using the Grocery Management System
```

Prints receipts and saves bill automatically:



```
GR

us

Date and time:23/12/2023 20:54:21['rice', 100,
TOTAL=1000
THANKS FOR SHOPPING WITH US
```



```
GROCERY STORE

user:admin

Date and time:23/12/2023 20:54:21
['rice', 100, 10]
TOTAL=1000
THANKS FOR SHOPPING WITH US
```

HARDWARE AND SOFTWARE REQUIREMENTS

I.OPERATING SYSTEM: Windows 7 or Above

Mac OS X 10.11 or higher, 64-bit

Linux: RHEL 6/7, 64-bit

II. PROCESSOR: Intel(R)Core (TMi5-10400F CPU @2.90GHz 2.90 GHz)

III. MOTHERBOARD: GIGABYTE B760M DS3H AX DDR4

IV. RAM :8 GB+

V. HARD DISK: SATA 40 GB OR ABOVE

VI. CD/DVD r/w multi drive combo: (If back up required)

VII. FLOPPY DRIVE: 1.44 MB (If Backup required)

VIII. MONITOR :15-17 inch

IX. Keyboard and mouse

X. Printer: Any good printer

SOFTWARE REQUIREMENTS:

I. Windows OS

II. Python IDLE

Installation procedure:

To install and subsequently use this,you need to have an application which could run .py files easily.

After that it is as simple as opening the file and using it as is.Although it doesn't have vibrance but we prioritized functionality so it is as optimized as a full scale up,while being light on storage.It is only 9kb ,it takes less storage than any flash game,and provides solutions to most of the problems a store owner could face while keeping and tallying all the data in a written way.

To ensure that the program runs smoothly we first have to enter all the items present in the shop, with thier price and quantity

After that the program handles the rest,from printing the bills to displaying all the current stocks

BIBLIOGRAPHY

1.Computer Science with Python :Sumita Arora
2.Websites: https://www.google.com
3.Websites: https://www.youtube.com