

**INFORMATICS PRACTICES**

**PROJECT FILE**

**ON**

**SHOP MANAGEMENT SYSTEM**



SAI ASWIN MADHAVAN

12 B

# INDEX

---

S.No	TOPIC	Page
1	AIM OF THE PROJECT	1
2	SYSTEM REQUIREMENTS	2
3	SYNOPSIS	3
4	SOURCE CODE	5
5	SAMPLE OUTPUT	23
6	DATA FILES	33
7	BIBLIOGRAPHY	36

# **1. AIM OF THE PROJECT**

---

**This project aims to provide start-ups, a tool to manage the stocks, and customer details, as well as to provide an in-built billing system. This project also aims to graphically interpret the growth in sales of the startup.**

## 2. SYSTEM REQUIREMENTS

---

- . Operating system: Windows, macOS, or Linux
- . RAM: 2 GB Minimum
- . Processor: Intel i3 (or equivalent) and above.
- . Storage: 2GB
- . Software required: Python, MySql

## 3. SYNOPSIS

---

### 3.1 ABOUT THE PROGRAM

It is a Shop Management System that allows small start-ups to manage the details of the available stocks and customer details. The program lets users add and edit items/stocks, record customer details, and a functional billing system. It also provides information about the sales of the shop in the form of graphs.

The program begins by asking for the login details of the user. The program also allows for creating a new account if required. A feature of this program is that it masks the password with ' \* ' while entering passwords and also hashes the password before saving the details in the Database.

The program allows user to:

1. View stock list
2. Edit stock list
  - a. Edit Item Name
  - b. Edit Item Cost
  - c. Edit Item Quantity
3. View customer list
4. Edit customer list
  - a. Edit customer Name
  - b. Edit customer DOB
  - c. Edit customer Phone Number
5. Generate bills for the customer
6. Sales Data Plots
  - a. Best Sold Item
  - b. Daily Return
  - c. Daily Items Sold

### 3.2 PROGRAM FILE STRUCTURE

Folder	File Name	Function Name
Shop Management	main	
	login	login
		create
		remove_acc
	item	view_item
		add_item
		remove_item
		edit_name
		edit_cost
		edit_qty
	cust	view_cust
		add_cust
		remove_cust
		edit_cust_name
		edit_cust_dob
		edit_cust_ph_no
	bill	bill
		dec_stock
		check_stock
	sales	insert_sales
		sales_plot
		insert_cust
		daily_return
		daily_sold

## 4. SOURCE CODE

---

### 4.1 MYSQL CODE

```
CREATE DATABASE SHOP;  
USE SHOP;
```

```
CREATE TABLE LOGIN (USERNAME VARCHAR  
(20) UNIQUE, PASSWORD VARCHAR(32));
```

```
CREATE TABLE ITEMS (SNO INT, ID CHAR(5) UNIQUE, ITEM_NAME  
VARCHAR(50), COST INT, QTY INT);
```

```
CREATE TABLE CUST (CUST_ID INT UNIQUE, CUST_NAME  
VARCHAR(50), DOB DATE, PH_NO INT)
```

```
CREATE TABLE SALES (ITEM_ID CHAR(5), ITEM  
VARCHAR(50), N_OF_PURCHASES INT);
```

```
CREATE TABLE CUST_VISIT (ID INT, NAME VARCHAR(50)  
VISITS INT);
```

```
CREATE TABLE ITEMLOG (DATE DATE, TIME, TIME, ITEM_ID  
CHAR(5), price int, quantity int, customer_id int);
```

## 4.2 PYTHON CODE

### 4.2.1 login.py

```
import mysql.connector

''' This python Module is to Manage Login Functions of the
project. '''

mycon = mysql.connector.connect(host = 'localhost',database =
'shop',user = 'root',passwd = '3333333')
c = mycon.cursor()

# Defining Login function
def login(usnm,pswd,s):

    c.execute('select password from login where username \
= "{}"'.format(usnm))

    records = c.fetchall()

    if records:
        c.execute('select md5("{}")'.format(pswd))
        pas = c.fetchall()[0][0]
        if pas == records[0][0]:
            s = True

        else:
            print('Incorrect username or password')
    else:
        print('Incorrect username or password')
    return(s)
```



```

# Define Create account function
def create(usnm,pswd):

    c.execute('select username from login where username \
= "{}"'.format(usnm))

    records = c.fetchall()
    if records:
        print('Username already exists')
        print('Please restart the program and try again')
        cs = False
    else:
        c.execute('insert into login\
values("{}","md5("{}")')'.format(usnm,pswd))
        mycon.commit()
        print('Successfully Created Account')
        print()
        cs = True
    return cs

# Define remove account function
def remove_acc(usnm):

    c.execute('delete from login where username \
= "{}"'.format(usnm))
    mycon.commit()

    print('Account Removed')

```

#### 4.2.2 item.py

```
import pandas as pd
import mysql.connector

''' This python module is to manage transactions regarding the items
table of the shop database. '''

mycon = mysql.connector.connect(host = 'localhost',\
database = 'shop',user = 'root',passwd = '3333333')
c = mycon.cursor()

def view_item(): # Function to view item list
    d = pd.read_sql_query('select * from items',mycon)
    print(d)

def add_item(item,cost,id,qty): # Function to add item to item list
    c.execute('select id from items where id = {}'.format(id))
    r = c.fetchall()
    if r:
        print('Id already exists. Please try again.')
    else:
        c.execute('select max(sno) from items')
        sno_max = c.fetchall()[0][0]
        if sno_max == None:
            sno = 1
        else:
            sno = sno_max + 1
        c.execute('insert into items values({}, "{}",\
                    "{}", {}, {})'.format(sno,id,item,cost,qty))
        print('Item added successfully')
    mycon.commit()

def remove_item(id): # Function to remove Item from list
    c.execute('delete from items where id = {}'.format(id))
    mycon.commit()

def edit_name(name,id): # Function to edit item name in list
    c.execute('update items set item_name = "{}"\
                where id = {}'.format(name,id))
    mycon.commit()
```

```
def edit_cost(cost,id): # Function to edit item cost in list
    c.execute('update items set cost = {} where id \
= {}'.format(cost,id))

    mycon.commit()

def edit_qty(qty,id): # Function to edit item quantity in list
    c.execute('update items set qty = {} where id \
= {}'.format(qty,id))

    mycon.commit()
```

#### 4.2.3 cust.py

```
import pandas as pd
import mysql.connector

''' This python module is to manage transactions
regarding the customer table of the shop database. '''

mycon = mysql.connector.connect(host = 'localhost',database \
= 'shop', user = 'root',passwd = '3333333')

c = mycon.cursor()

def view_cust(): # Function to view customer list
    d = pd.read_sql_query('select * from cust',mycon)
    print(d)

def add_cust(customer,dob,ph_no): # Function to add customer to list
    c.execute('select max(cust_id) from cust')
    id_max = c.fetchall()[0][0]

    if id_max == None:
        id = 1
    else:
        id = id_max+1

    c.execute('select cust_id from cust where cust_id \
= {}'.format(id))

    r = c.fetchall()
    if r:
        print('Id already exists. Please try again.')
    else:
        c.execute('insert into cust values("{}","{}",\
        "{}",{})'.format(id,customer,dob,ph_no))
        print('Customer added successfully')
    mycon.commit()
```

```
def remove_cust(id): # function to remove customer from list
    c.execute('delete from cust where cust_id = {}'.format(id))
    mycon.commit()
```

```
def edit_cust_name(name,id): # Edit a customer's name in list
    c.execute('update cust set cust_name = "{}"\
              where cust_id = {}'.format(name,id))
    mycon.commit()
```

```
def edit_cust_dob(dob,id): # Edit customer Date of birth in list
    c.execute('update cust set dob = "{}" where cust_id \
= {}'.format(dob,id))

    mycon.commit()
```

```
def edit_cust_ph_no(ph_no,id): # Edit Customer Phone number in list
    c.execute('update cust set ph_no = {}where cust_id \
= {}'.format(ph_no,id))

    mycon.commit()
```

#### 4.2.4 sales.py

```
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

''' This python module is to manage sales and customer graphs
and updates transactions and customer details. '''

mycon = mysql.connector.connect(host = 'localhost',database \
= 'shop',user = 'root',passwd = '3333333')

c = mycon.cursor()

def insert_sales(id,q,cid): # Function to update sales table after
every purchase

    c.execute('select cost from items where id = {}'.format(id))
    cost = c.fetchall()[0][0]
    c.execute('insert into itemlog values(curdate(),\
        curtime(),{}, {}, {}, {})' .format(id,cost,q,cid))

    # Query to check if item already exists
    c.execute('select item_id from sales where item_id \
= {}'.format(id))

    sales_id = c.fetchall()

    if sales_id: # if item exists, update sales table

        c.execute('select n_of_purchases from sales \
            where item_id = {}'.format(id))
        quan = c.fetchall()[0][0]
        quan += q

        c.execute('update sales set n_of_purchases = {} \
            where item_id = "{}"'.format(quan,id))
        mycon.commit()
```

```

        else: # else, insert a new item to sales table
            c.execute('select item_name from items where id \
= {}'.format(id))

            item = c.fetchall()[0][0]

            c.execute('insert into sales values\
            ("{}", "{}", {})'.format(id, item, q))
            mycon.commit()

def sales_plot(): # Plot a graph against items and number of
purchases
    df = pd.read_sql_query('select item,n_of_purchases from\
sales',mycon)
    x = df['item']
    y = df['n_of_purchases']

    plt.bar(x,y)
    plt.xticks(rotation = 30)

    plt.xlabel('Items')
    plt.ylabel('No. of Purchases')
    plt.show()

def insert_cust(id): # Function to update sales table after every
purchase
    c.execute('select id from cust_visit where id \
= {}'.format(id))

    cust_id = c.fetchall()

    if cust_id: # If customer exists in table, update the number
of visits

        c.execute('select visits from cust_visit\
            where id = {}'.format(id))
        visit = c.fetchall()[0][0]
        visit += 1
        c.execute('update cust_visit set visits = {} \
            where id = "{}"'.format(visit, id))
        mycon.commit()

```

```

else: # else, insert customer details into table

    c.execute('select cust_name from cust \
              where cust_id = {}'.format(id))
    item = c.fetchall()[0][0]

    c.execute('insert into cust_visit \
              values("{}","{}",{})'.format(id,item,1))
    mycon.commit()

def daily_return(): # Plot a graph against Day and amount earned
that day
    df = pd.read_sql_query('select Date,sum(price*quantity)\
                          "Daily Return" from itemlog group\
                          by date order by date',mycon)
    x = [str(i) for i in df['Date']]
    y = df['Daily Return']

    plt.plot(x,y)
    plt.xticks(rotation = 30)
    plt.xlabel('Date')
    plt.ylabel('Income')
    plt.show()

def daily_sold(): # Plot a graph against Day and number of items
sold that day
    df = pd.read_sql_query('select Date,sum(quantity)\
                          "Daily Return" from itemlog\
                          group by date order by date',mycon)
    x = [str(i) for i in df['Date']]
    y = df['Daily Return']

    plt.plot(x,y)
    plt.xticks(rotation = 30)
    plt.xlabel('Date')
    plt.ylabel('Items')
    plt.show()

```



#### 4.2.5 `bill.py`

```
import mysql.connector
import pandas as pd

''' This python module is to manage the billing system of the
software. '''

mycon = mysql.connector.connect(host = 'localhost',\
database = 'shop',user = 'root',passwd = '3333333')
c = mycon.cursor()

def bill(cid,i_dic):# Define function bill
    c.execute('select cust_name from cust where \
               cust_id = "{}".format(cid)) # accessing customer
name
    name = c.fetchall()[0][0]
    lid = sorted(i_dic)
    q = [i_dic[i] for i in lid]

    n = len(lid)
    l = str(tuple(lid)) # formatted list of item_id to use in
query
    if n == 1:
        query = f'select id,item_name,cost from \
                 items where id in ({lid[0]})'
    else:
        query = f'select id,item_name,cost from items where id\
in {l}'

    df = pd.read_sql_query(query,mycon) # Create dataframe from
sql query
    df['Quantity'] = q
    df = df.rename(columns =
{'id':'ID','item_name':'Items','cost':'Cost'})

    sum_list = (df['Cost'])*(df['Quantity']) # Price = Cost x
Quantity
    price = sum_list.sum() # Calculating total price
```

```

c.execute("select now()")
dt = c.fetchall()[0][0]

# Printing the bill
print('-----')
print('|                                BILL                                |')
print('-----')
print(f'Customer ID: {cid}                                Name: {name}')
print(f'Date/Time: {dt}')
print('-----')
print(df)
print('-----')
print(f'Total Price :                                AED {price}')
print('-----')
print('-----')

# Define function dec_stock to decrease the quantity in stock from
sql table after billing
def dec_stock(i_list,i_q):
    for i in range(len(i_list)):
        c.execute('select qty from items where id\
= {}'.format(i_list[i]))
        r = c.fetchall()[0][0]
        c.execute('update items set qty = {} \
                    where id = {}'.format(r-i_q[i],i_list[i]))
        mycon.commit()

def check_stock(id):
    c.execute('select qty from items where id = {}'.format(id))
    stock = c.fetchall()[0][0]
    return stock

```

#### 4.2.6 main.py

```
# importing required modules
import mysql.connector
import login as lo
import item as it
import cust as cu
import bill as bi
import sales as sa
import stdiomask

print('\nShop Management System')
print('-----\n')

# creating necessary variables
state = False
create_state = True

# Start Program
print('To login Enter 1,\nTo create new account Enter 2.')

create_acc = input('Enter here : ')

if create_acc == '2':
    usr = input('Enter New Username (Max: 20 characters) : ')
    passwd = stdiomask.getpass('Enter New Password : ')
    print()
    create_state = lo.create(usr,passwd)# Create account function
    create_acc = '1'

if create_state == True and create_acc == '1':
    usnm = input('Enter Username : ')
    pswd = stdiomask.getpass('Enter Password : ')
    state = lo.login(usnm,pswd,state)# login function
```

```

if state: # If signed in ...
    print('Successfully signed in')
    print()
    continuing = True
    while continuing: # This is the main loop of the program
        print('To stop the program, Enter 1')
        print('To remove account and end program, Enter 2')
        print('To view item list, Enter 3')
        print('To edit item list, Enter 4')
        print('To view customer list, Enter 5')
        print('To edit customer list, Enter 6')
        print('To bill and end program, Enter 7')
        print('To view sales graph, Enter 8')
        print('To view daily returns, Enter 9')
        print('To view daily items sold, Enter 10')
        print()
        next = input('Enter Here : ')
        print()

        if next == '1':
            continuing = False # stop the program

        elif next == '2':
            lo.remove_acc(usnm) # Remove_account function from
module login.py

            print()
            continuing = False # stop the program

        elif next == '3':
            it.view_item() # function to view item list from
module item.py

            print()

        elif next == '4':
            print('To add item, Enter 1')
            print('To remove item, Enter 2')
            print('To edit an item, Enter 3')
            print()
            e_list = input('Enter here : ') # accept input for
editing item list

```

```

print()
if e_list == '1':
    item = input('Enter name of item : ')
    cost = input('Enter cost of item : ')
    id = input('Enter 5 digit item ID : ')
    qty = int(input('Enter quantity of item : '))
    it.add_item(item, cost, id, qty) # add_item function
    print()
elif e_list == '2':
    id = input('Enter ID of item to remove : ')
    it.remove_item(id) # remove_item function from
module item.py

    print('Item successfully removed')
    print()

elif e_list == '3':
    id = input('Enter ID of item to edit : ')
    print('To edit name of item, Enter 1')
    print('To edit cost of item, Enter 2')
    print('To edit quantity of item, Enter 3')
    edit = input('Enter here : ')
    if edit == '1':
        name = input('Enter new name : ')
        it.edit_name(name, id) # edit_name function
        print()
    elif edit == '2':
        cost = input('Enter new cost : ')
        it.edit_cost(cost, id) # edit_cost function
        print()
    elif edit == '3':
        qty = input('Enter new quantity : ')
        it.edit_qty(qty, id) # edit_quantity function
        print()

elif next == '5':
    cu.view_cust() # Function to view customer list from
module item.py

    print()

```

```

elif next == '6':
    print('To add customer, Enter 1')
    print('To remove customer, Enter 2')
    print('To edit an customer, Enter 3')
    print()
    c_list = input('Enter here : ') # accept input for
editing cust list

    print()
    if c_list == '1':
        customer = input('Enter name of customer : ')
        dob = input('Enter Date of birth of customer(YYYY\
-MM-DD) : ')

        ph_no = int(input('Enter 10 digit phone number \
: '))

        cu.add_cust(customer,dob,ph_no) # function to add
new customer

        print()
        elif c_list == '2':
            id = input('Enter ID of customer to remove : ')
            cu.remove_cust(id) # function to remove customer
            print('Customer successfully removed')
            print()
            elif c_list == '3':
                id = input('Enter ID of customer to edit : ')
                print('To edit name of customer, Enter 1')
                print('To edit date of birth of customer, Enter\
2')

                print('To edit phone number of customer, Enter 3')
                edit = input('Enter here : ')

                if edit == '1':
                    name = input('Enter new name : ')
                    cu.edit_cust_name(name,id) # function to edit
customer name

                    print()
                    elif edit == '2':
                        dob = input('Enter new date of birth(YYYY-MM-
DD) : ')

```

```

        cu.edit_cust_dob(dob,id) # function to edit
customer dob
        print()
        elif edit == '3':
            ph_no = input('Enter new phone number : ')
            cu.edit_cust_ph_no(ph_no,id) #function to edit
customer p.no
        print()

        elif next == '7': # billing system
            visit = input('Is customer visiting for the first \
time ? (y/n) : ')

            print()
            if visit == 'y':
                print('Please add customer to list before \
billing')
                print()
            elif visit == 'n':
                print('Item List')
                it.view_item()
                print()
                cid = input('Enter customer ID : ')
                sa.insert_cust(cid)
                print()
                i_list = []
                i_q = []
                i_dic = {}
                billing = True
                while billing: # Generate item list for billing
                    lid = input(f'Enter ID of item : ')
                    q = int(input('Enter quantity of item : '))
                    if q <= bi.check_stock(lid):
                        print()
                        i_list.append(lid)
                        i_q.append(q)
                        i_dic[lid] = q
                        sa.insert_sales(lid,q,cid)

```

```

        else:
            print('Required quantity of stock \
unavailable..\n')
            qn = input('Do you want to continue ? (y/n) :
')

            if qn == 'n':
                billing = False

            print()
            bi.bill(cid,i_dic)
            bi.dec_stock(i_list,i_q)
            print()
            continuing = False

    elif next == '8':
        sa.sales_plot() # Sales plot function from sales.py

    elif next == '9':
        sa.daily_return() # Daily Return function from
sales.py

    elif next == '10':
        sa.daily_sold() # Daily items sold function from
sales.py

    print('Program successfully Ended')

cu.mycon.close()
it.mycon.close()
bi.mycon.close()
sa.mycon.close()
lo.mycon.close()

```



## 5. SAMPLE OUTPUT

---

First of all, the User logs in

```
Shop Management System
-----
To login Enter 1,
To create new account Enter 2.
Enter here : █
```

**LOGIN** – If you already have a login id

**CREATE A NEW ACCOUNT** – If you don't have a login id

In this case we will sign up,

```
To login Enter 1,
To create new account Enter 2.
Enter here : 2
Enter New Username (Max: 20 characters) : admin
Enter New Password : ***** → admin
Successfully Created Account
```

Now we will login,

```
Enter Username : admin
Enter Password : ***** → admin
Successfully signed in
```

After logging in,

```
To stop the program, Enter 1  
To remove account and end program, Enter 2  
To view item list, Enter 3  
To edit item list, Enter 4  
To view customer list, Enter 5  
To edit customer list, Enter 6  
To bill and end program, Enter 7  
To view sales graph, Enter 8  
To view daily returns, Enter 9  
To view daily items sold, Enter 10
```

Enter Here :

1. To stop the program.

```
Enter Here : 1
```

```
Program successfully Ended
```

2. To remove current account.

```
Enter Here : 2
```

```
Account Removed
```

```
Program successfully Ended
```

3. To view item list.

Enter Here : 3

	SNo	id	Item_name	Cost	Qty
0	1	00001	Laptop	3000	91
1	2	00002	Desktop	2500	98
2	3	00003	CPU	1000	97
3	4	00004	Printer	500	97
4	5	00005	Headphones	250	94
5	6	00006	Earphones	100	96
6	7	00007	Mouse	50	98
7	8	00008	Keyboard	75	98
8	9	00009	Webcam	100	91
9	10	00010	Mobile Phone	2000	95
10	11	00011	Television	3500	97
11	12	00012	Gaming Console	1500	94
12	13	00013	Speaker	500	83
13	14	00014	Lamp	200	96
14	15	00015	Calculator	50	95

4. To edit item list.

Enter Here : 4

To add item, Enter 1

To remove item, Enter 2

To edit an item, Enter 3

Enter here : ■

a. Add Item.

Enter here : 1

Enter name of item : Tablet

Enter cost of item : 1000

Enter 5 digit item ID : 00016

Enter quantity of item : 100

Item added successfully

	SNo	id	Item_name	Cost	Qty
0	1	00001	Laptop	3000	91
1	2	00002	Desktop	2500	98
2	3	00003	CPU	1000	97
3	4	00004	Printer	500	97
4	5	00005	Headphones	250	94
5	6	00006	Earphones	100	96
6	7	00007	Mouse	50	98
7	8	00008	Keyboard	75	98
8	9	00009	Webcam	100	91
9	10	00010	Mobile Phone	2000	95
10	11	00011	Television	3500	97
11	12	00012	Gaming Console	1500	94
12	13	00013	Speaker	500	83
13	14	00014	Lamp	200	96
14	15	00015	Calculator	50	95
15	16	00016	Tablet	1000	100

- b. To edit an item. Example: To change cost of Tablet to 1500

```

Enter here : 3

Enter ID of item to edit : 00016
To edit name of item, Enter 1
To edit cost of item, Enter 2
To edit quantity of item, Enter 3
Enter here : 2
Enter new cost : 1500

```

	SNo	id	Item_name	Cost	Qty
0	1	00001	Laptop	3000	91
1	2	00002	Desktop	2500	98
2	3	00003	CPU	1000	97
3	4	00004	Printer	500	97
4	5	00005	Headphones	250	94
5	6	00006	Earphones	100	96
6	7	00007	Mouse	50	98
7	8	00008	Keyboard	75	98
8	9	00009	Webcam	100	91
9	10	00010	Mobile Phone	2000	95
10	11	00011	Television	3500	97
11	12	00012	Gaming Console	1500	94
12	13	00013	Speaker	500	83
13	14	00014	Lamp	200	96
14	15	00015	Calculator	50	95
15	16	00016	Tablet	1500	100

- c. To remove an item, say tablet.

Enter here : 2

Enter ID of item to remove : 00016

Item successfully removed

	SNo	id	Item_name	Cost	Qty
0	1	00001	Laptop	3000	91
1	2	00002	Desktop	2500	98
2	3	00003	CPU	1000	97
3	4	00004	Printer	500	97
4	5	00005	Headphones	250	94
5	6	00006	Earphones	100	96
6	7	00007	Mouse	50	98
7	8	00008	Keyboard	75	98
8	9	00009	Webcam	100	91
9	10	00010	Mobile Phone	2000	95
10	11	00011	Television	3500	97
11	12	00012	Gaming Console	1500	94
12	13	00013	Speaker	500	83
13	14	00014	Lamp	200	96
14	15	00015	Calculator	50	95

5. To view customer list.

Enter Here : 5

	cust_id	cust_name	dob	ph_no
0	1	Rajiv	2003-12-12	501231234
1	2	Manoj	2002-01-12	551323454
2	3	Abood	2000-02-23	501523654
3	4	Raj	2001-12-27	506973654
4	5	Riya	2002-06-17	500925654
5	6	Dean	1999-06-20	501295814
6	7	Vidya	1999-08-22	501274665
7	8	Peeyush	1997-09-02	501435431
8	9	Sunil	2004-03-21	501093745
9	10	Pandya	2001-11-30	551638667

6. To edit customer list.

```
Enter Here : 6

To add customer, Enter 1
To remove customer, Enter 2
To edit an customer, Enter 3

Enter here : █
```

a. To add a customer.

```
Enter here : 1

Enter name of customer : Rohan
Enter Date of birth of customer(YYYY-MM-DD) : 2001-03-27
Enter 10 digit phone number : 0505478734
Customer added successfully
```

	cust_id	cust_name	dob	ph_no
0	1	Rajiv	2003-12-12	501231234
1	2	Manoj	2002-01-12	551323454
2	3	Abood	2000-02-23	501523654
3	4	Raj	2001-12-27	506973654
4	5	Riya	2002-06-17	500925654
5	6	Dean	1999-06-20	501295814
6	7	Vidya	1999-08-22	501274665
7	8	Peeyush	1997-09-02	501435431
8	9	Sunil	2004-03-21	501093745
9	10	Pandya	2001-11-30	551638667
10	11	Rohan	2001-03-27	505478734

b. To Edit customer details, Example: Change name of Rohan to Rahman.

```
Enter here : 3

Enter ID of customer to edit : 11
To edit name of customer, Enter 1
To edit date of birth of customer, Enter 2
To edit phone number of customer, Enter 3
Enter here : 1
Enter new name : Rahman
```

	cust_id	cust_name	dob	ph_no
0	1	Rajiv	2003-12-12	501231234
1	2	Manoj	2002-01-12	551323454
2	3	Abood	2000-02-23	501523654
3	4	Raj	2001-12-27	506973654
4	5	Riya	2002-06-17	500925654
5	6	Dean	1999-06-20	501295814
6	7	Vidya	1999-08-22	501274665
7	8	Peeyush	1997-09-02	501435431
8	9	Sunil	2004-03-21	501093745
9	10	Pandya	2001-11-30	551638667
10	11	Rahman	2001-03-27	505478734

c. To Remove a customer, Say Rahman.

```
Enter here : 2

Enter ID of customer to remove : 11
Customer successfully removed
```

	cust_id	cust_name	dob	ph_no
0	1	Rajiv	2003-12-12	501231234
1	2	Manoj	2002-01-12	551323454
2	3	Abood	2000-02-23	501523654
3	4	Raj	2001-12-27	506973654
4	5	Riya	2002-06-17	500925654
5	6	Dean	1999-06-20	501295814
6	7	Vidya	1999-08-22	501274665
7	8	Peeyush	1997-09-02	501435431
8	9	Sunil	2004-03-21	501093745
9	10	Pandya	2001-11-30	551638667

7. To use the billing system.

Let's use the billing system to bill for customer 'Sunil', Who has ordered a Laptop, two Headphones, and a Speaker.

```
Enter Here : 7

Is customer visiting for the first time ? (y/n) : n

Item List
  SNo    id    Item_name    Cost    Qty
0      1  00001      Laptop    3000    91
1      2  00002      Desktop    2500    98
2      3  00003         CPU    1000    97
3      4  00004      Printer     500    97
4      5  00005    Headphones     250    94
5      6  00006    Earphones     100    96
6      7  00007        Mouse      50    98
7      8  00008      Keyboard      75    98
8      9  00009        Webcam     100    91
9     10  00010    Mobile Phone    2000    95
10     11  00011      Television    3500    97
11     12  00012  Gaming Console    1500    94
12     13  00013        Speaker     500    83
13     14  00014         Lamp      200    96
14     15  00015    Calculator      50    95
```

```
Enter customer ID : 9

Enter ID of item : 00001
Enter quantity of item : 1

Do you want to continue ? (y/n) : y
Enter ID of item : 00005
Enter quantity of item : 2

Do you want to continue ? (y/n) : y
Enter ID of item : 00013
Enter quantity of item : 1

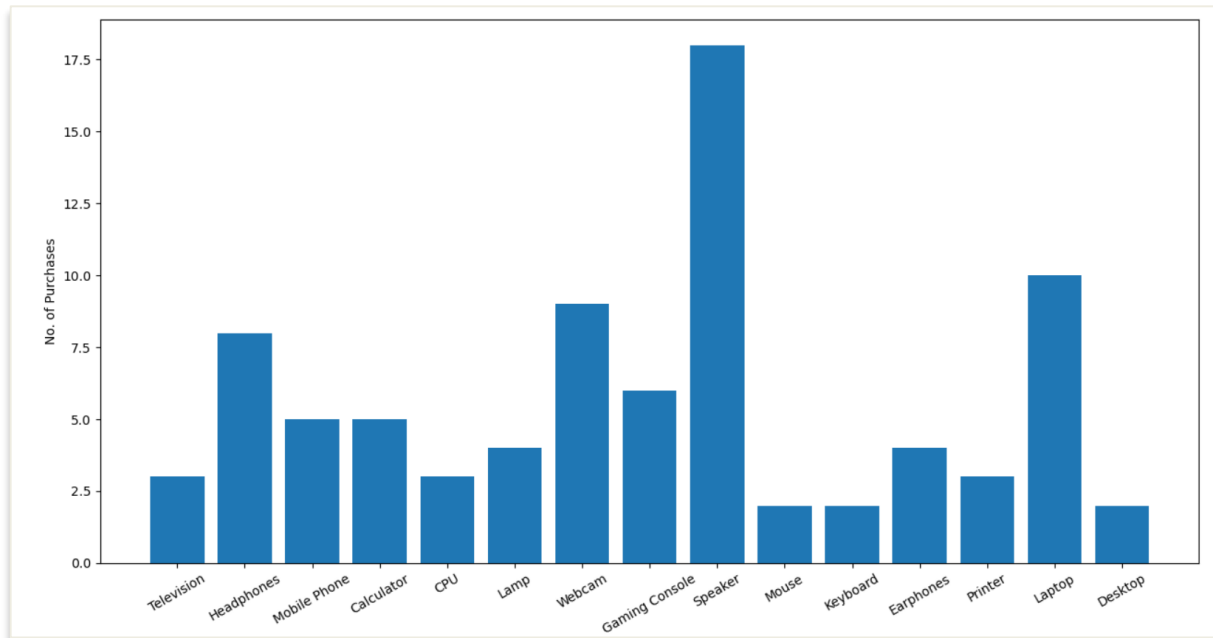
Do you want to continue ? (y/n) : n
```

```
-----
|                               BILL                               |
|-----|
Customer ID: 9                Name: Sunil
Date/Time: 2020-12-13 16:56:50
|-----|
  ID    Items    Cost    Quantity
0 00001    Laptop    3000         1
1 00005  Headphones     250         2
2 00013    Speaker     500         1
|-----|
Total Price :                AED 4000
|-----|
|-----|
```



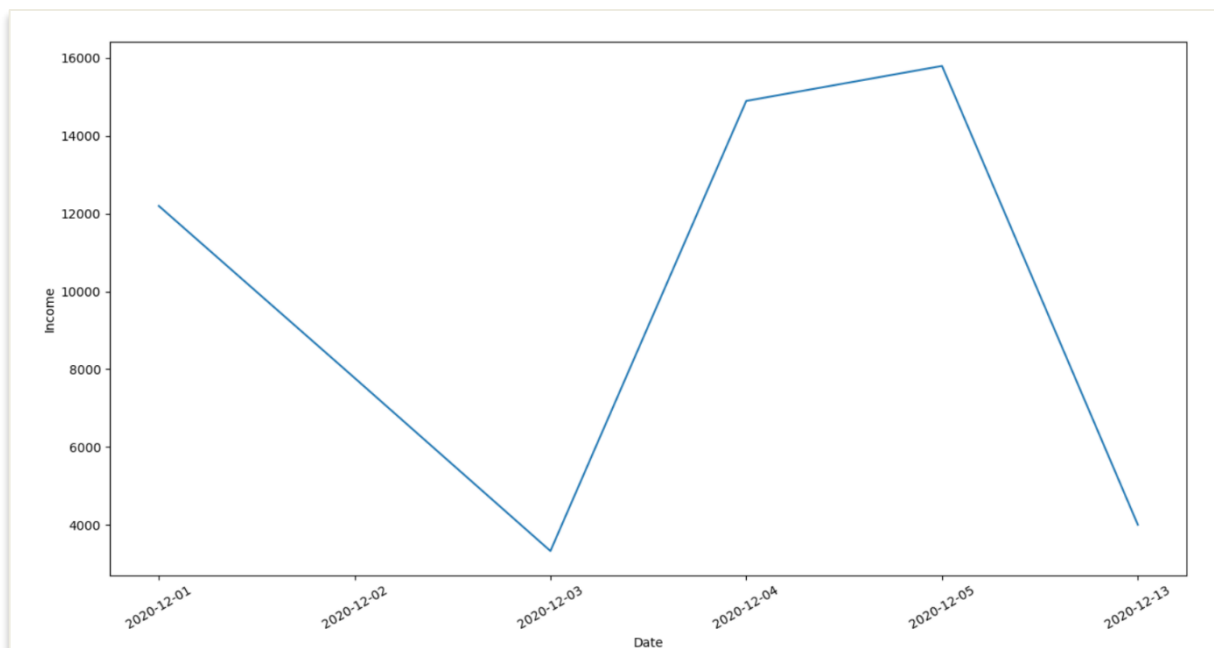
8. To view sales graph.

Enter Here : 8



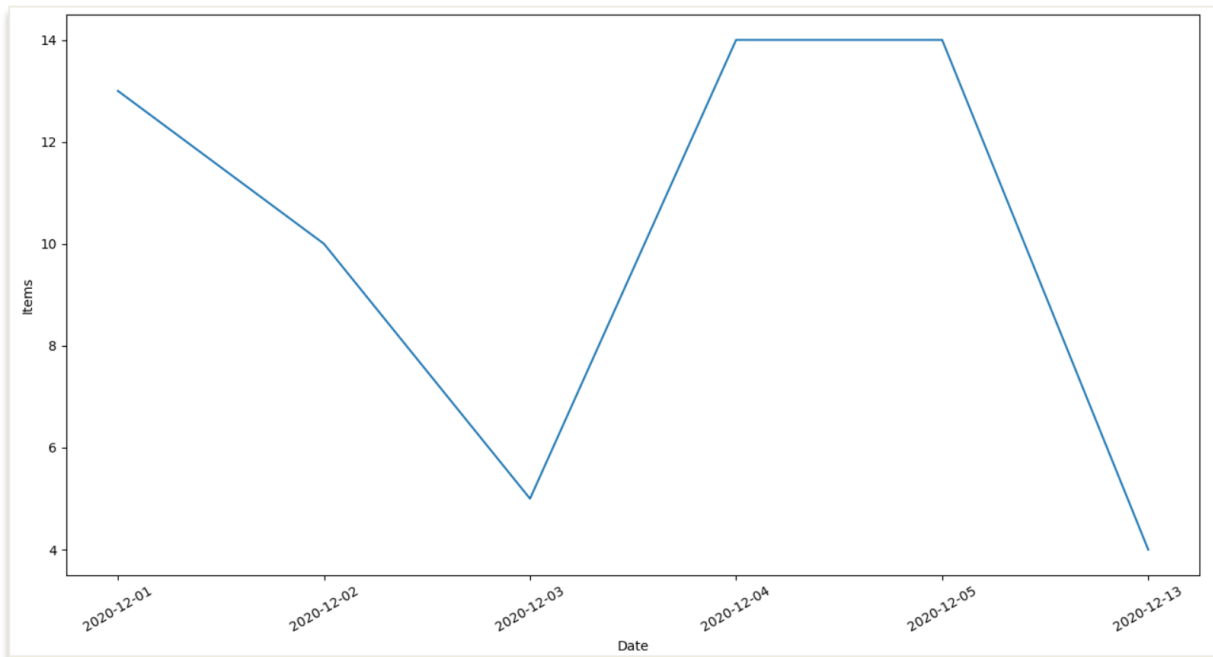
9. To view daily income graph.

Enter Here : 9



10. To view daily goods sold graph.

Enter Here : 10



## 6. DATAFILES

Database : **SHOP**

Tables Used :

Table 1 : **LOGIN**

Field	Type	Null	Key	Default	Extra
username	varchar(20)	YES	UNI	NULL	
password	varchar(32)	YES		NULL	

username	password
admin	21232f297a57a5a743894a0e4a801fc3

Table 2 : **ITEMS**

Field	Type	Null	Key	Default	Extra
sno	int	YES		NULL	
id	char(5)	YES	UNI	NULL	
item_name	varchar(50)	YES		NULL	
cost	int	YES		NULL	
qty	int	YES		NULL	

SNo	id	Item_name	Cost	Qty
1	00001	Laptop	3000	90
2	00002	Desktop	2500	98
3	00003	CPU	1000	97
4	00004	Printer	500	97
5	00005	Headphones	250	92
6	00006	Earphones	100	96
7	00007	Mouse	50	98
8	00008	Keyboard	75	98
9	00009	Webcam	100	91
10	00010	Mobile Phone	2000	95
11	00011	Television	3500	97
12	00012	Gaming Console	1500	94
13	00013	Speaker	500	82
14	00014	Lamp	200	96
15	00015	Calculator	50	95

Table 3 : CUST

Field	Type	Null	Key	Default	Extra
cust_id	int	YES	UNI	NULL	
cust_name	varchar(50)	YES		NULL	
dob	date	YES		NULL	
ph_no	int	YES		NULL	

cust_id	cust_name	dob	ph_no
1	Rajiv	2003-12-12	501231234
2	Manoj	2002-01-12	551323454
3	Abood	2000-02-23	501523654
4	Raj	2001-12-27	506973654
5	Riya	2002-06-17	500925654
6	Dean	1999-06-20	501295814
7	Vidya	1999-08-22	501274665
8	Peeyush	1997-09-02	501435431
9	Sunil	2004-03-21	501093745
10	Pandya	2001-11-30	551638667

Table 4 : SALES

Field	Type	Null	Key	Default	Extra
item_id	char(5)	YES		NULL	
item	varchar(50)	YES		NULL	
n_of_purchases	int	YES		NULL	

item_id	Item	n_of_purchases
00011	Television	3
00005	Headphones	8
00010	Mobile Phone	5
00015	Calculator	5
00003	CPU	3
00014	Lamp	4
00009	Webcam	9
00012	Gaming Console	6
00013	Speaker	18
00007	Mouse	2
00008	Keyboard	2
00006	Earphones	4
00004	Printer	3
00001	Laptop	10
00002	Desktop	2

Table 5 : CUST\_VISIT

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
name	varchar(50)	YES		NULL	
visits	int	YES		NULL	

id	name	visits
5	Riya	2
1	Rajiv	2
10	Pandya	2
4	Raj	2
6	Dean	2
9	Sunil	3
8	Peeyush	2
3	Abood	1
7	Vidya	7
2	Manoj	1

Table 6 : ITEMLOG (5 sample records shown )

Field	Type	Null	Key	Default	Extra
Date	date	YES		NULL	
Time	time	YES		NULL	
Item_id	char(5)	YES		NULL	
Price	int	YES		NULL	
Quantity	int	YES		NULL	
Customer_id	int	YES		NULL	

Date	Time	Item_id	Price	Quantity	Customer_id
2020-12-05	08:36:15	00010	2000	1	10
2020-12-05	08:36:24	00006	100	1	10
2020-12-05	08:37:34	00001	3000	1	4
2020-12-05	08:37:44	00005	250	1	4
2020-12-05	08:38:38	00012	1500	1	9

## 7. BIBLIOGRAPHY

---

- Book: Informatics Practices by Sumita Arora
- Websites:
  - <https://stackoverflow.com>
  - <https://w3schools.com>
  - <https://www.geeksforgeeks.org>
  - <https://w3resource.com>
  - <https://youtube.com>
  - <http://www.vecteezy.com>