**Dr. Amrit Lal Ishrat Memorial Sunbeam School Chaubeypur**

**DALIMSS SUNBEAM CHAUBEYPUR**

**Affiliated to CBSE-(10+2) New** Delhi

AISSCE : 2022-23

Project Report

on

**INVENTORY MANAGEMENT SYSTEM**

**Submitted To**

**Mr. Vidhu Mishra**

**DALIMSS SUNBEAM SCHOOL CHAUBEYPUR**

**Submitted By**

**Name : Kishan Chaubey**

**Class : XII Science**

**Roll No.: \_\_\_\_\_\_\_\_\_**

## DECLARATION

I **KISHAN CHAUBEY**, bearing roll no \_\_\_\_\_\_\_\_\_\_, a student of Class XII *The* **DALIMSS SUNBEAM SCHOOL CHUABEYPUR**hereby declare that I own the full responsibility for the information, results etc. provided in this PROJECT titled “**Inventory Management System**”. It has been developed successfully by using the Data Handling concepts like data management, data visualization etc. provided in the programming language Python at The **DALIMSS SUNBEAM SCHOOL CHUABEYPUR** in complete fulfillment of project (curriculum of Central Board of Secondary Education CBSE of Informatics Practices (065) conducted by CBSE, New Delhi for the academic session 2022-23.

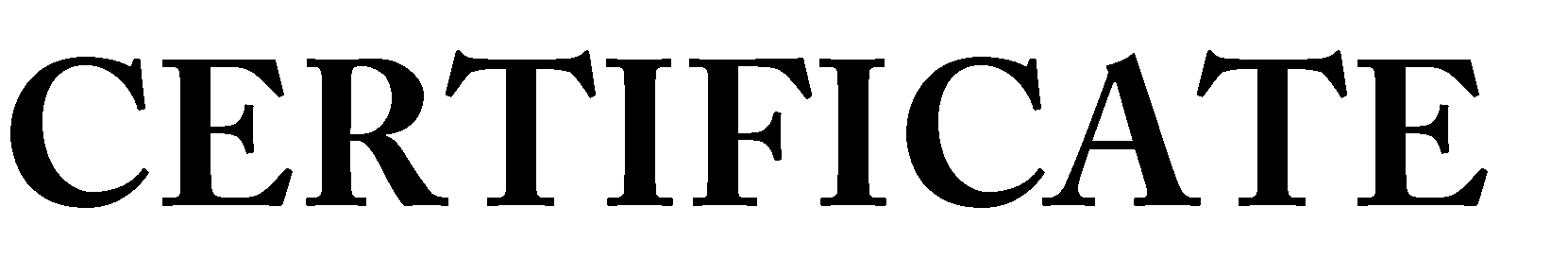
I also declare that this project work has neither been submitted to any other board nor published at any time by me in the past.

KISHAN CHAUBEY

**Roll No:**

**Class: XII**

**DALIMSS SUNBEAM SCHOOL CHUABEYPUR**



*This is to certify that the Informatics Practices project on* ***Inventory Management System*** *has been successfully completed by* ***KISHAN CHAUBEY*** *of* ***Class XII,* DALIMSS SUNBEAM SCHOOL CHUABEYPUR** *for consideration in partial fulfillment of curriculum of Central Board of Secondary Education (CBSE) of Informatics Practices (065) for the award of AISSCE Practical Examination 2022-23.*

*I certify that this project is up to my expectation and as per the guidelines issued by the CBSE.*

**(External Examiner)**

**(Internal Examiner)** **(Principal)**

## ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude to all those who have been instrumental in preparation of this project.

I feel great pleasure to express my obligation to Mr. TARUN RUPANI, Principal of **DALIMSS SUNBEAM SCHOOL CHUABEYPUR**.

I am also sincerely grateful to Mr. **VIDHU MISHRA** PGT (Computer Science), The **DALIMSS SUNBEAM SCHOOL CHUABEYPUR** for his encouragement and valuable guidance during the entire period of work.

I would also thank all of my parents and friends for their whole hearted support and encouragement without with this project would not have been successful.

I could not forget Internet, Textbooks which provided me with sufficient matter for reference.

## TABLE OF CONTENT

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Topic** | **Page No** |
|  | Introduction | 6 |
|  | Problem Statement | 7 |
|  | Objective | 8 |
|  | Project Scope | 9 |
|  | System Requirement and Specification | 10 |
|  | Overview of Python | 11 |
|  | Overview of SQL | 12 |
|  | Project Module | 13 |
|  | Data Flow Diagram | 14 |
|  | Database Design and Table Structure | 15 |
|  | SQL Code | 16 |
|  | Python Code | 19 |
|  | Outputs | 32 |
|  | Data Analysis (Best Selling Product) | 39 |
|  | Conclusion | 41 |
|  | Bibliography | 42 |

**INTRODUCTION**

Inventory refers to all the goods, items and materials purchased or manufactured by business for sale to the customer to make profit.

Inventory management is all about tracking and controlling of business inventory right from manufacturing, buying to storing and using. It controls the entire flow of goods from purchasing to sale and ensures that you always have the right quantities of the right item in the right location at the right time.

Inventory Management System is an application which refers to Inventory Management developed for small business. It can be used by business to manage Inventory using a computerized system where they can manage details of purchase, sale, products and customers. They can also analyze data by visualization.

## PROBLEM STATEMENT

For any business Inventory is one of the most important department that must be well managed in order to run daily business activity smoothly. But mostly business are not able to manage inventory as they do not have good computerized system. As a result they lacks in

* Security of data, documents related to business transactions
* Unable to keep single copy of data at single location which increases data complexity
* Managing data consistency
* Accessing and referencing or search data quickly
* Unable to analyze data as there is no mechanism to visualize data

## OBJECTIVE

The objective of project is to build an application program to:

* Provide function to manage goods in the store more efficiently.
* Provide searching facility based on various factors.
* Reduce time and cost to control and manage inventory.
* Reduce paperwork.
* Increased accuracy and reliability.
* Increased Data Consistency.
* Increased Data security.

This application program can be used easily by non-programming personals.

## PROJECT SCOPE

Managing all products, sufficient stocks, sale and purchase records and analyzing sales is a tedious job for any business. To do it more effectively and correctly a good Inventory control is required. This is provided by our application which have following scope:

* ensures effective inventory control
* manage daily sales records easily and precisely
* can perform sales analysis of various products
* user friendly interface
* very easy to use
* High level security of data.

**SYSTEM REQUIREMENT AND SPECIFICATIONS**

**Software Requirements:**

|  |  |
| --- | --- |
| **Operating System** | Window-7 and later versions (32bit, 64 bit) |
| **Language** | Python |
| **Plateform** | Python IDLE 3.7 (min) |
| **Database** | MySQL |
| **Database driver** | MySQL Connector |
| **Plotting** | Matplotlib |

**Hardware Requirements:**

|  |  |
| --- | --- |
| **Processor** | Pentium Dual Core (min) 32bit or 64 bit |
| **Hard-Disk** | 160GB (min) |
| **RAM** | 1GB (min) |

**Input/output Requirements:**

|  |  |
| --- | --- |
| **Input** | Mouse (any) |
| **Input** | Keyboard (any) |
| **Output** | Monitor (any) |
| **Output** | Printer (any) |

## OVERVIEW OF PYTHON

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. Guido Van Rossum is known as the founder of Python programming.

**Features of Python:**

* Python is a high level language. It is a free and open source language.
* It is an interpreted language, as Python programs are executed by an interpreter.
* Python programs are easy to understand as they have a clearly defined syntax and relatively simple structure.
* Python is case-sensitive. For TECHTIPNOW, NUMBER and number are not same in Python.
* Python is portable and platform independent, means it can run on various operating systems and hardware platforms.
* Python has a rich library of predefined functions.
* Python is also helpful in web development. Many popular web services and applications are built using Python.
* Python uses indentation for blocks and nested blocks.

## OVERVIEW OF MYSQL

MySQL is a relational database management system (RDBMS). It is pronounced as "My Sequel”. MySQL was originally founded and developed in Sweden by David Axmark, Allan Larsson and Michael Widenius, who had worked together since the 1980s.

**Characteristics of MySQL:**

* MySQL is released under an open-source license so it is customizable.
* It requires no cost or payment for its usage.
* MySQL has superior speed, is easy to use and is reliable.
* MySQL uses a standard form of the well-known ANSI-SQL standards.
* MySQL is a platform independent application which works on many operating systems like
* Windows, UNIX, LINUX etc. and have compatibility with many languages including JAVA, C++, PHP, PERL, etc.
* MySQL is an easy to install RDBMS and is capable of handling large data sets.

Since MySQL is released under an open-source license, it does not require any cost or payment for its usage. Anyone can download and use this software from specific location on Internet.

## PROJECT MODULES

Inventory Management System is divided into following modules:

**Customer module:** This module helps you to add, modify, and delete customer data.

**Supplier module:** This module helps you to add, modify, and delete supplier data.

**Stock module:** This module helps you to add, modify, and delete product data.

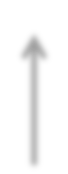
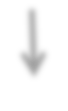
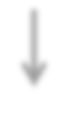
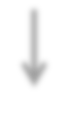
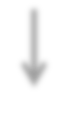
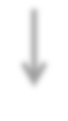
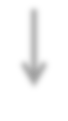
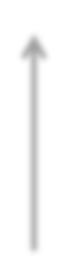
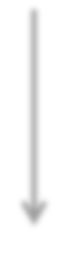
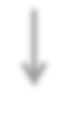
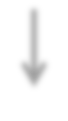
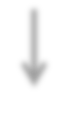
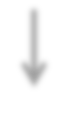
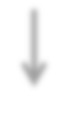
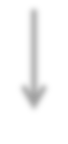
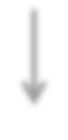
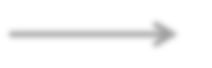
**Sale module:** This module is used to manage sell information. **Purchase module:** This module is used to manage purchase records. **Login Module:** This module is used to manage login details.

**User module:** This module is used to manage user credentials.

**Report module:** This module is used to generate sale bills, purchase bills and visualize data by plotting charts.

## DATA FLOW DIAGRAM

### PYTHON INTERFACE



**ADMIN**

**LOGIN**

d

**fail**

**LOGIN CHECK**

**Success**

**MENU**

**SALE**

**PURCHASE**

**STOCK**

**CUSTOMER**

**SUPPLIER**

**UDATE**

**ADD**

**SEARCH**

**DELETE**

**MySQL Data Store**

**DATABASE DESIGN AND TABLE STRUCTUR**

|  |  |
| --- | --- |
| **Item** | |
| INO | INT (PRIMARY KEY) |
| INAME | VARCHAR |
| PRATE | FLOAT |
| SRATE | FLOAT |
| QOH | INT |

|  |  |
| --- | --- |
| **SMASTER** | |
| SALEID | INT (PRIMARY KEY) |
| SDATE | DATE |
| SID | INT (FOREIGN KEY) |
| TOTAL | FLOAT |

|  |  |
| --- | --- |
| **CUSTOMER** | |
| CID | INT (PRIMARY KEY) |
| CNAME | VARCHAR |
| CADD | VARCHAR |
| MOBILE | CHAR(10) |

|  |  |
| --- | --- |
| **SDETAILS** | |
| SALEID | INT (FOREIGN KEY) |
| INO | INT(FOREIGN KEY) |
| QTY | INT |
| RATE | FLOAT |
| TOTAL | FLOAT |

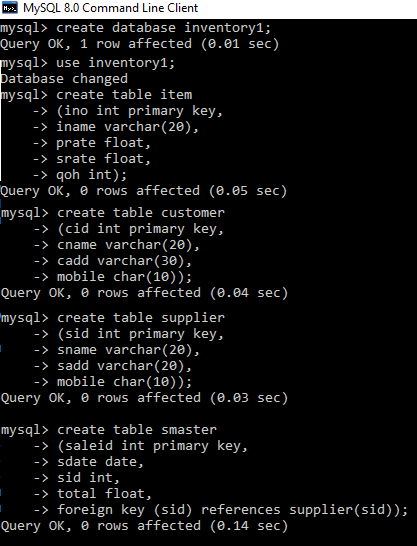
|  |  |
| --- | --- |
| **SUPPLIER** | |
| SID | INT (PRIMARY KEY) |
| SNAME | VARCHAR |
| SADD | VARCHAR |
| MOBILE | CHAR(10) |

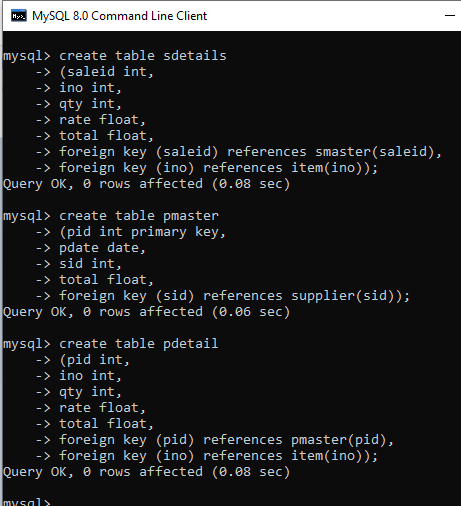
|  |  |
| --- | --- |
| **PDETAILS** | |
| PID | INT (FOREIGN KEY) |
| INO | INT(FOREIGN KEY) |
| QTY | INT |
| RATE | FLOAT |
| TOTAL | FLOAT |

|  |  |
| --- | --- |
| **PMASTER** | |
| PID | INT (PRIMARY KEY) |
| PDATE | DATE |
| SID | INT (FOREIGN KEY) |
| TOTAL | FLOAT |

# SQL CODE

**DATABASE DESIGN AND TABLE STRUCTURES**





# PYTHON SOURCE CODE

## MODULE DESIGNS

**MAIN MODULE**

import item import customer import supplier import transaction import report import os while(True):

os.system('cls')

print("="\*80)

print("-"\*80)

print("\n\t\t\tInventory Management System\n") print("-"\*80)

print("="\*80)

print("\n\t\t\t\tEnter your choice\n\t\t\t\t1.Items\n\t\t\t\

\t2.Customers\n\t\t\t\t3.Suppliers\

\n\t\t\t\t4.Transaction\n\t\t\t\t5.Report\n\t\t\t\t6.Exit") ch=int(input())

if ch==1: while(True):

print("---------------\nEnter your choice\n \

\n\t\t\t\t1.Add Item\n\t\t\t\t2.Edit Item\n\t\t\t\t3.Fix Rate\

\n\t\t\t\t4.Search Item\n\t\t\t\t5.Delete Item\n\t\t\t\t6.Exit") ch=int(input())

if ch==1: item.add\_item()

elif ch==2: item.edit\_item()

elif ch==3: item.fix\_rate()

elif ch==4: item.search\_item()

elif ch==5: item.delete\_item()

elif ch==6: break

elif ch==2: while(True):

print("\n---------------\nEnter your choice\n \

\n\t\t\t\t1.Add customers\n\t\t\t\t2.Edit Customers\

\n\t\t\t\t3.Search Customers\n\t\t\t\t4.Delete Customers\n\t\t\t\t5.exit") ch=int(input())

if ch==1: customer.add\_customer()

elif ch==2: customer.edit\_customer()

elif ch==3: customer.search\_customer()

elif ch==4: customer.delete\_customer()

elif ch==5: break

elif ch==3: while(True):

print("\n---------------\nEnter your choice\n \

\n\t\t\t\t1.Add Suppliers\n\t\t\t\t2.Edit Suppliers\

\n\t\t\t\t3.Search Suppliers\n\t\t\t\t4.Delete Suppliers\n\t\t\t\t5.exit")

ch=int(input()) if ch==1:

supplier.add\_supplier() elif ch==2:

supplier.edit\_supplier() elif ch==3:

supplier.search\_supplier() elif ch==4:

supplier.delete\_supplier() elif ch==5:

break elif ch==4:

while(True):

print("\n---------------\nEnter your choice\n \

\n\t\t\t\t1.Sale\n\t\t\t\t2.Purchase\n\t\t\t\t3.exit") ch=int(input())

if ch==1: transaction.sale()

elif ch==2: transaction.purchase()

elif ch==3: break

elif ch==5: while(True):

print("\n---------------\nEnter your choice\n \

\n\t\t\t\t1.Item Details\n\t\t\t\t2.Customer Details\

\n\t\t\t\t3.Supplier Details\n\t\t\t\t4.Sale Details\n\

\t\t\t5.Purchase Details\n\t\t\t\t6.Best Selling Product(Plot)\

\n\t\t\t\t7.Sale Performance(Plot)\n\t\t\t\t8.exit") ch=int(input())

if ch==1:

report.show\_item() elif ch==2:

report.show\_customer() elif ch==3:

report.show\_supplier() elif ch==4:

report.show\_sale() elif ch==5:

report.show\_purchase() elif ch==6:

report.best\_product() elif ch==7:

report.sale\_performance() elif ch==8:

break elif ch==6:

break

import pandas as pd

## ITEM MODULE

from tabulate import tabulate import mysql.connector as sqlt import matplotlib.pyplot as plt

con=sqlt.connect(host = "localhost", user = "root", passwd="sanjay", database = "inventory")

cursor=con.cursor() def add\_item():

try:

ino = int(input("Enter Item No")) iname = input("Enter Iname")

prate=float(input("Enter Purchase Rate")) srate=float(input("enter Sale Rate")) qoh=int(input("Enter Qty On Hand"))

q="insert into item values({},'{}',{},{},{});".format(ino,iname,prate,srate,qoh) cursor.execute(q)

con.commit() print("Item Added")

except:

print("Wrong Entry..Please check") def edit\_item():

try:

ino=int(input("Enter Item No"))

q="select \* from item where ino = {};".format(ino) cursor.execute(q)

if cursor.fetchone(): iname=input("Enter Item Name")

cursor.execute("update item set iname = '{}' where ino={};".format(iname,ino)) con.commit()

print("item Edited") else:

print("Item Not Found") except:

print("Wrong Entry") def fix\_rate():

ino=int(input("Enter Item No"))

q="select \* from item where ino = {};".format(ino) cursor.execute(q)

if cursor.fetchone():

prate=int(input("enter new purchase rate")) srate=int(input("Enter new Sale rate"))

cursor.execute("update item set prate={},srate={} where ino={};".format(prate,srate,ino))

con.commit()

print("New rate applied") else:

print("Item Not Found") def search\_item():

ino=int(input("Enter Item No"))

q="select \* from item where ino = {};".format(ino) cursor.execute(q)

if cursor.fetchone(): df=pd.read\_sql(q,con)

print(tabulate(df,headers="keys", tablefmt = "psql", showindex = False)) else:

print("Item Not Found") def delete\_item():

ino=int(input("Enter Item No"))

q="select \* from item where ino = {};".format(ino) cursor.execute(q)

if cursor.fetchone():

cursor.execute("delete from item where ino={};".format(ino)) con.commit()

print("item deleted") else:

print("Item Not Found")

## CUSTOMER MODULE

import pandas as pd

from tabulate import tabulate import mysql.connector as sqlt import matplotlib.pyplot as plt

con=sqlt.connect(host = "localhost", user = "root", passwd="sanjay", database = "inventory") cursor=con.cursor()

def add\_customer():

cid = int(input("Enter Customer ID")) cname = input("Enter Customer Name") cadd=input("Enter Address") mobile=input("Enter Mobile")

q="insert into customer values({},'{}','{}','{}');".format(cid,cname,cadd,mobile) cursor.execute(q)

con.commit() print("Customer Added")

def edit\_customer(): cid=int(input("Enter Customer ID"))

q="select \* from Customer where cid = {};".format(cid) cursor.execute(q)

if cursor.fetchone():

cadd=input("Enter Customer Address")

cursor.execute("update customer set cadd = '{}' where cid={};".format(cadd,cid)) con.commit()

print("Customer Edited") else:

print("Customer Not Found") def search\_customer():

cname=input("Enter Customer Name")

q="select \* from customer where cname like '%{}%';".format(cname) cursor.execute(q)

if cursor.fetchall(): df=pd.read\_sql(q,con)

print(tabulate(df,headers='keys',tablefmt='psql',showindex=False)) else:

print("Customer Not found") def delete\_customer():

cid=int(input("Enter Customer ID"))

q="select \* from customer where cid = {};".format(cid) cursor.execute(q)

if cursor.fetchone():

cursor.execute("delete from customer where cid={};".format(cid)) con.commit()

print("customer deleted") else:

print("customer Not Found")

## SUPPLIER MODULE

import pandas as pd

from tabulate import tabulate import mysql.connector as sqlt import matplotlib.pyplot as plt

con=sqlt.connect(host = "localhost", user = "root", passwd="sanjay", database = "inventory") cursor=con.cursor()

def add\_supplier():

sid = int(input("Enter Supplier ID")) sname = input("Enter Supplier Name") sadd=input("Enter Address") mobile=input("Enter Mobile")

q="insert into supplier values({},'{}','{}','{}');".format(sid,sname,sadd,mobile) cursor.execute(q)

con.commit() print("Supplier Added")

def edit\_supplier(): sid=int(input("Enter Supplier ID"))

q="select \* from Supplier where sid = {};".format(sid) cursor.execute(q)

if cursor.fetchone():

sadd=input("Enter Supplier Address")

cursor.execute("update Supplier set sadd = '{}' where sid={};".format(sadd,sid)) con.commit()

print("Supplier Edited") else:

print("Supplier Not Found") def search\_supplier():

sid=int(input("Enter Supplier ID"))

q="select \* from Supplier where sid = {};".format(sid) cursor.execute(q)

if cursor.fetchone(): df=pd.read\_sql(q,con)

print(tabulate(df,headers="keys", tablefmt = "psql", showindex = False)) else:

print("Supplier Not Found") def delete\_supplier():

sid=int(input("Enter Supplier ID"))

q="select \* from Supplier where sid = {};".format(sid) cursor.execute(q)

if cursor.fetchone():

cursor.execute("delete from Supplier where sid={};".format(sid)) con.commit()

print("Supplier deleted") else:

print("Supplier Not Found")

## TRANSACTION MODULE

import pandas as pd

from tabulate import tabulate import mysql.connector as sqlt import matplotlib.pyplot as plt

con=sqlt.connect(host = "localhost", user = "root", passwd="sanjay", database = "inventory")

cursor=con.cursor() def purchase():

pid=0 total=0 grand=0 l=[]

ch='y'

q="select max(pid) as largest from pmaster" cursor.execute(q)

r=cursor.fetchone()[0] if r:

pid=r+1 else:

pid=1

pdate=input("Enter Purchase date") sid = int(input("Enter Supplier ID"))

cursor.execute("select \* from supplier where sid={};".format(sid)) if cursor.fetchone():

print("Item Details") df=pd.read\_sql("select \* from item",con)

print(tabulate(df,headers='keys',tablefmt='psql',showindex=False)) while(ch=='y'):

ino=int(input("Enter Item No"))

cursor.execute("select \* from item where ino ={};".format(ino)) r1=cursor.fetchone()

if r1:

qty = int(input("Enter qty")) rate=r1[2]

total=qty\*rate grand=grand+total t=(pid,ino,qty,rate,total) l.append(t)

else:

print("Item Not Found")

ch=input("Do you wish to add more Items in bucket y/n")

q1="insert into pmaster values({},'{}',{},{});".format(pid,pdate,sid,grand) cursor.execute(q1)

con.commit()

q2="insert into pdetail values(%s,%s,%s,%s,%s);" cursor.executemany(q2,l)

con.commit()

cursor.executemany("insert into ptemp values(%s,%s,%s,%s,%s);",l) con.commit()

q3="update item join ptemp using(ino) set item.qoh = item.qoh+ptemp.qty" cursor.execute(q3)

con.commit()

cursor.execute("delete from ptemp") con.commit()

print("Item Purchased and Added") else:

print("Supplier Not Found") def sale():

saleid=0

total=0 grand=0 l=[]

ch='y'

q="select max(saleid) as largest from smaster" cursor.execute(q)

r=cursor.fetchone()[0] if r:

saleid=r+1 else:

saleid=1

sdate=input("Enter Sale date")

sid = int(input("Enter Supplier ID"))

cursor.execute("select \* from supplier where sid={};".format(sid)) if cursor.fetchone():

print("Item Details") df=pd.read\_sql("select \* from item",con)

print(tabulate(df,headers='keys',tablefmt='psql',showindex=False)) while(ch=='y'):

ino=int(input("Enter Item No"))

cursor.execute("select \* from item where ino ={};".format(ino)) r1=cursor.fetchone()

if r1:

qty = int(input("Enter qty")) rate=r1[2]

total=qty\*rate grand=grand+total

t=(saleid,ino,qty,rate,total) l.append(t)

else:

print("Item Not Found")

ch=input("Do you wish to add more Items in bucket y/n")

q1="insert into smaster values({},'{}',{},{});".format(saleid,sdate,sid,grand) cursor.execute(q1)

con.commit()

q2="insert into sdetail values(%s,%s,%s,%s,%s);" cursor.executemany(q2,l)

con.commit()

cursor.executemany("insert into stemp values(%s,%s,%s,%s,%s);",l) con.commit()

q3="update item join stemp using(ino) set item.qoh = item.qoh-stemp.qty" cursor.execute(q3)

con.commit()

cursor.execute("delete from stemp") con.commit()

print("Item Purchased and Added") else:

print("Supplier Not Found")

## REPORT MODULE

import mysql.connector as sqlt import pandas as pd

from tabulate import tabulate

con = sqlt.connect(host = "localhost", user = "root", passwd = "sanjay", database = "library")

cursor = con.cursor() def show\_item():

df=pd.read\_sql("select \* from item",con)

print(tabulate(df,headers= 'keys', tablefmt='psql',showindex = False)) def show\_customer():

df=pd.read\_sql("select \* from customer",con) print(tabulate(df,headers= 'keys', tablefmt='psql',showindex = False))

def show\_supplier():

df=pd.read\_sql("select \* from supplier",con) print(tabulate(df,headers= 'keys', tablefmt='psql',showindex = False))

def show\_sale():

bdate=input("enter beginning date") edate=input("enter end date")

df=pd.read\_sql("select \* from smaster where sdate between '{}' and '{}';".format(bdate,edate),con)

print(tabulate(df,headers= 'keys', tablefmt='psql',showindex = False)) def show\_purchase():

bdate=input("enter beginning date") edate=input("enter end date")

df=pd.read\_sql("select \* from pmaster where pdate between '{}' and '{}';".format(bdate,edate),con)

print(tabulate(df,headers= 'keys', tablefmt='psql',showindex = False)) def best\_product():

s=input("Enter Start date") e=input("Enter End Date")

q="select s2.ino,sum(s2.qty) as total from smaster s1,sdetail s2 \ where s1.saleid = s2.saleid and s1.sdate between '{}' and '{}'\ group by s2.ino;".format(s,e)

df=pd.read\_sql(q,con)

print(tabulate(df, headers='keys', tablefmt = 'psql', showindex=False)) plt.bar(df.ino,df.total)

plt.xlabel("Item Code") plt.ylabel("Qty")

plt.title("Best Selling Product") plt.xticks(df.ino)

plt.show()

def sale\_performance():

y=input("Enter Year")

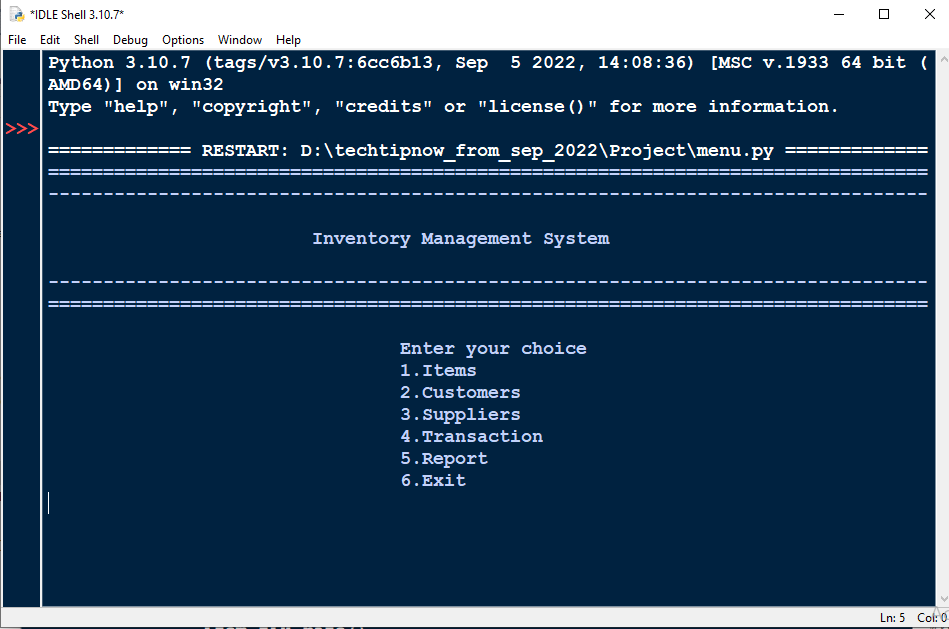
q="select month(sdate) as month,sum(total)\ as total from smaster where year(sdate) = '{}'\ and group by month(sdate);".format(y)

df = pd.read\_sql(q,con)

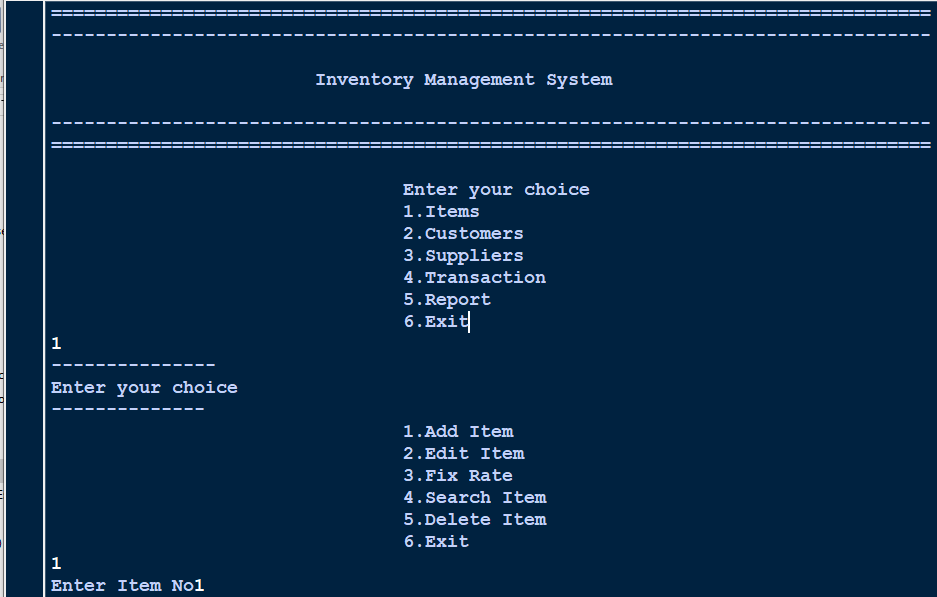
plt.plot(df.month,df.total) plt.xlabel("Month") plt.ylabel("Total Sale") plt.xticks(df.month) plt.show()

# PROGRAM OUTPUT

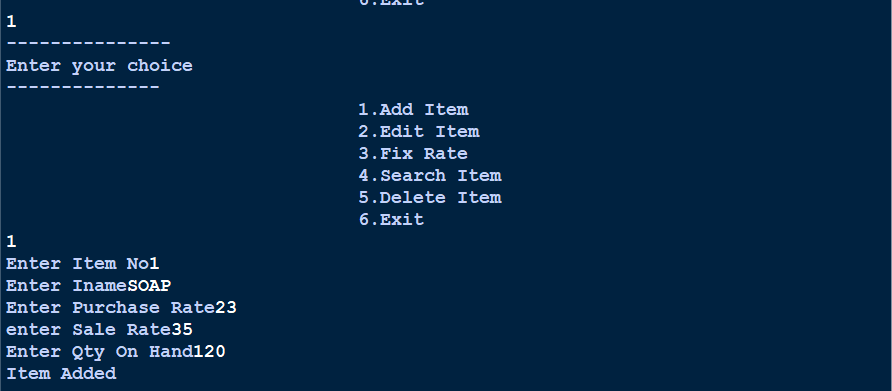
## MAIN MENU



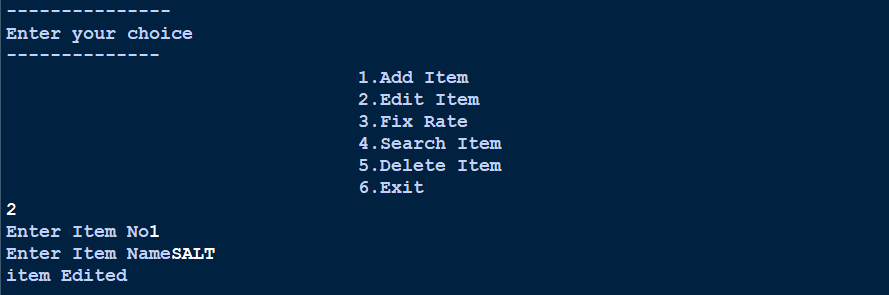
**ITEM**



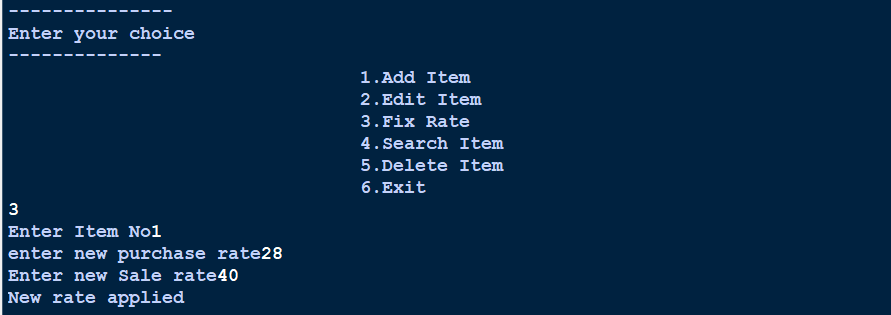
### ADD ITEM



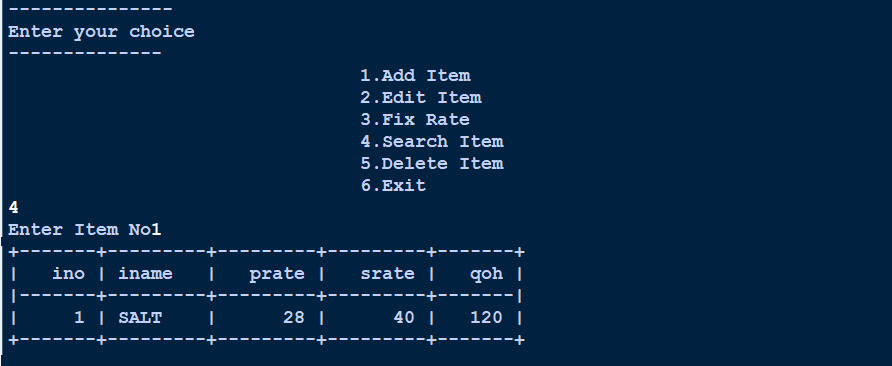
**EDIT ITEM**



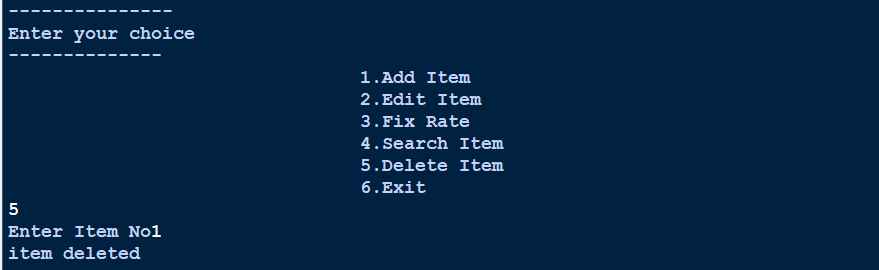
### UPDATE RATE OF ITEM



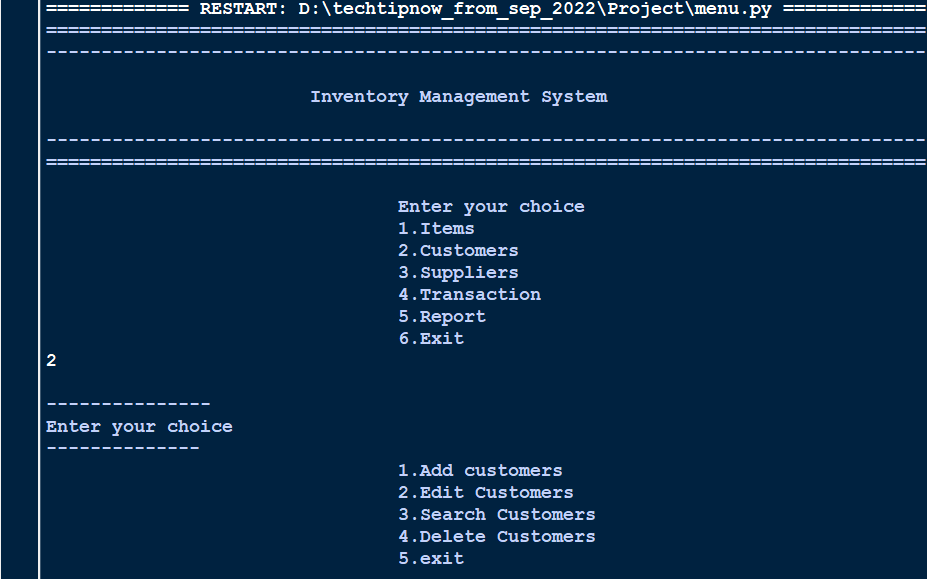
**SEARCH ITEM**



**DELETE ITEM**



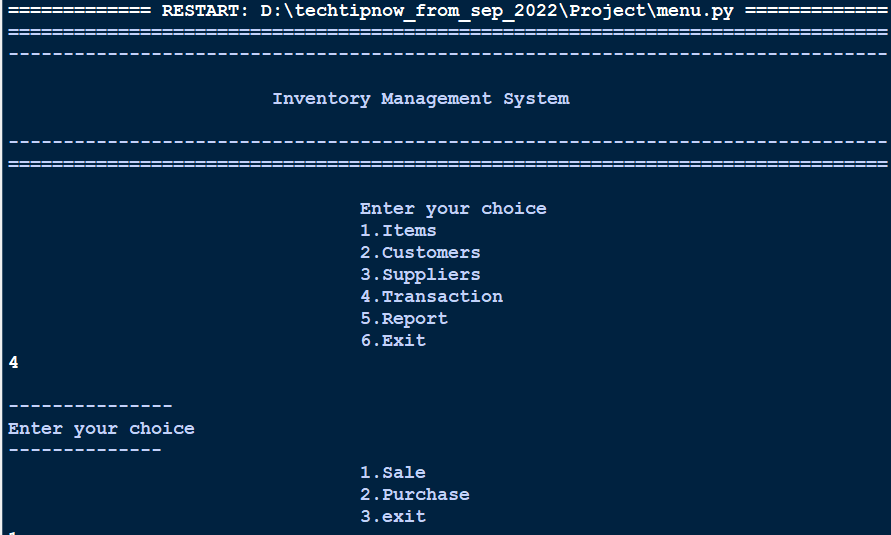
## CUSTOMER MODULE



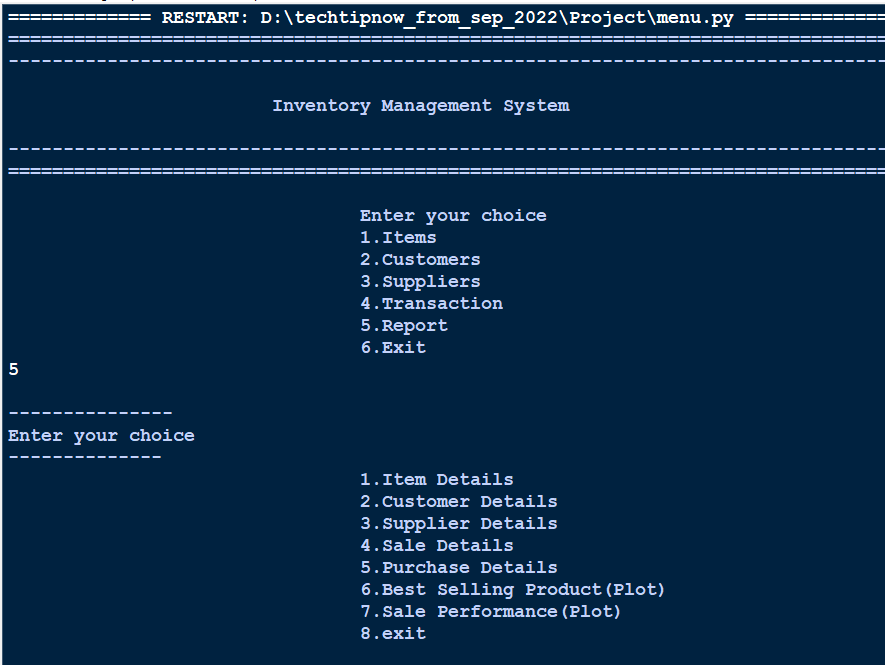
**SUPPLIER MODULE**



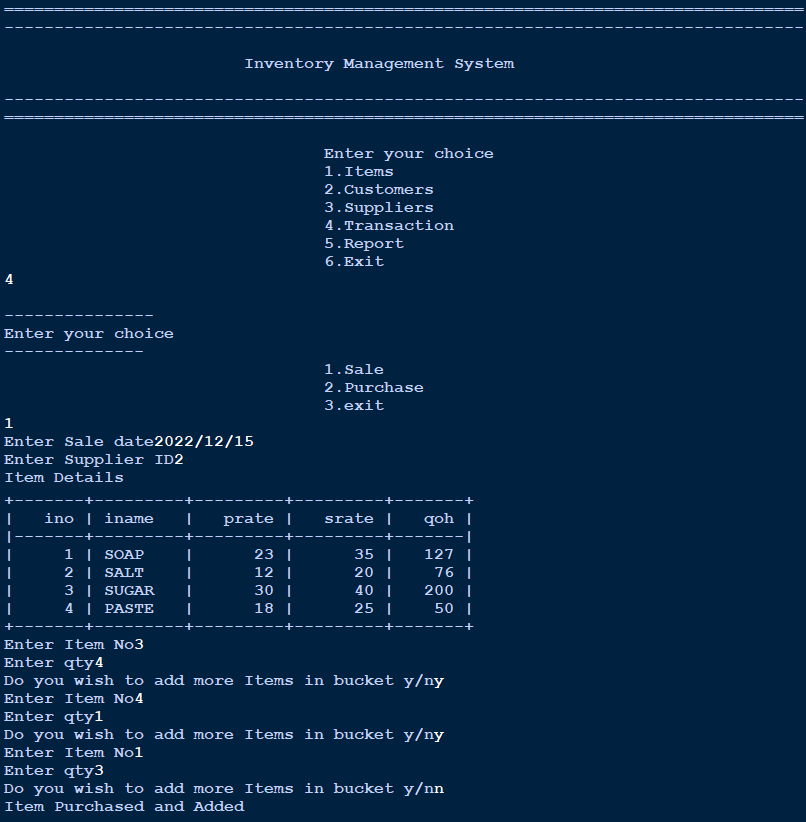
## TRANSACTION MODULE



**REPORT MODULE**



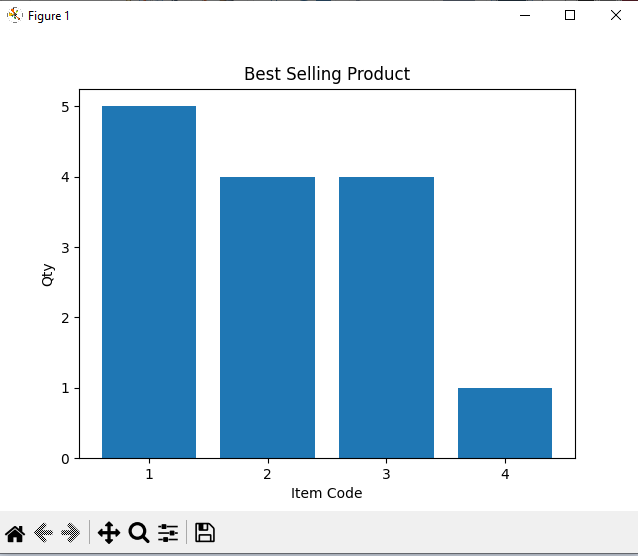
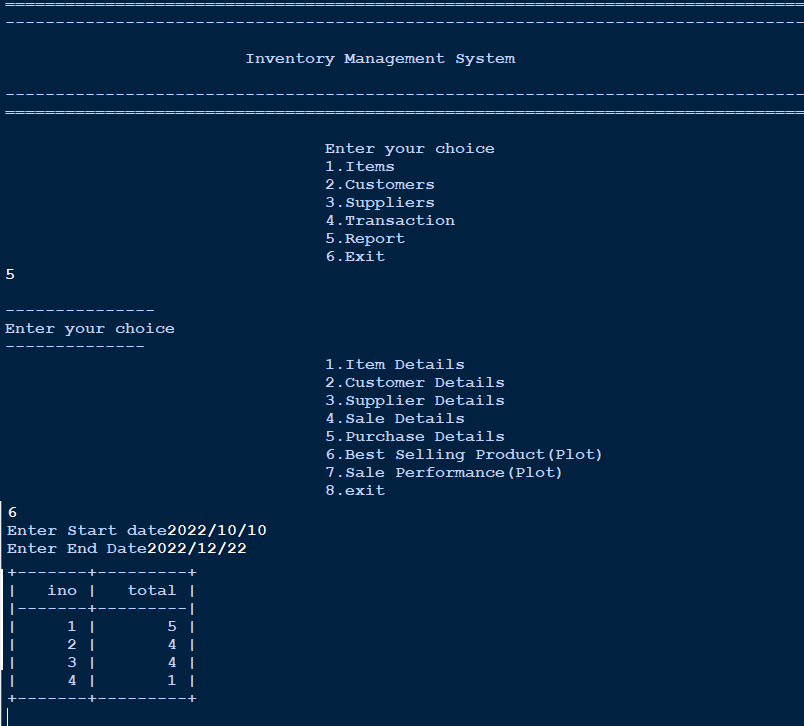
**SALE MODULE**



# DATA ANALYSIS

**BEST SELLING PRODUCT**

## BEST SELLING PRODUCT



**CONCLUSION**

This Inventory Management System is a simple desktop based application basically suitable for small organization. It has all basic elements which are used for managing inventory of small organization. We are successful in making the application where we can insert, delete, update, search and analyze records as per need. This application also provides a report including chart of sales records to analyze the performance.

We strongly believes that he implementation of this system will surely benefit the organization.

## BIBLIOGRAPHY

#### References and Bibliography:

* Informatics Practices Class-XII NCERT Publication
* Informatics Practices Class-XII by Sumita Arora
* Think Python by Allen B Downey
* Python for everybody by Charles Severance
* Learning MYSQL by Seyed M. M. Tahaghoghi
* MySQL in a nutshell by Russell J. T. Dyer
* www.google.com
* [www.ncert.co.in](http://www.ncert.co.in/)
* etc.