Project Report

On

stock management system

p.g.t.(comp. science)

submitted by;

name:- AGRASTH NAMan

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr. no. | Topic | Page no. |
| 1 | Introduction | 5 |
| 2 | Objective | 6 |
| 4 | Working of the project | 7 |
| 5 | Libraries used | 13 |
| 6 | MySql tables | 14 |
| 7 | Source code | 17 |
| 8 | Output | 46 |
| 9 | Technology used | 59 |
| 10 | Bibliography | 60 |

**INTRODUCTION**

Stock management system is the computerized application to automate the activities in buying and selling stocks.

The main aim of this project is to manage the stocks in a wholesale shop. This project is helpful for maintaining the record of purchase and sale.

This project has the type of system which provides the owner of the wholesale shop to manage the records without any difficulty.

Proper comments have been given at desired locations to make the project user friendly. Various functions and structures are used to make complete use of the programming language.

**OBJECTIVE**

The objectives of doing this project are the following:

* To enhance the ability to critically analyze a problem and to design implement and evaluate a computing solution that meets requirements.
* An ability to use current tools and methodologies in computing practice.
* The ability to design and implement system that involves hardware and software and an interaction between the two.
* The project is mainly focused to apply the concepts of python learnt in class XII as per the syllabus prescribed by CBSE

**WORKING OF THE PROJECT**

(Note- A database named stock should be already created )

The project contains following modules:-

1. Product Management: This module is used to add, update and delete the products.

2. Purchase Management: This module is used to manage the purchase system.

3. Sales Management: This module is used to manage the sale of the products.

4. User Management: This module is used to add/delete the user/staff.

5. Database setup: This module is used to setup the database in the system for the first time.

* Options and their task:

1. NEW STOCK

2. OLD STOCK

1. If new start is selected, you will get a list of options which are as follows :

1. CREATE A NEW DATABASE

2. OPEN MAIN MENU

3. BACK

Option 1 will create a new database for the system and will open the main menu and option 2 will directly open the main menu. If you directly open the menu you can access the whole system but not the user management option, to access it you will have to enter the management password. Option 3 will bring you back to page 1.

1. If continue stock is selected, you will be asked for your user id and password. Once you have entered them correctly, the system will provide you with the main menu for the further management work.

(Note- the user id and password entered here should already be enrolled in the user management option.)

* MAIN MENU:

1. PRODUCT MANAGEMENT

2. PURCHASE MANAGEMENT

3. SALES MANAGEMENT

4. USER MANAGEMENT

5. DATABASE SETUP

6. EXIT

7. CLOSE PC

1. If product management is selected,you will get a list of options which is as follows :

1. Add New Product

2. List Product

3. Update Product

4. Delete Product

5. Back (Main Menu)

Option 1 will provide you with an option to add any product in your stock. Option 2 will provide you with the list of products in your stock with the options such as, 1. List all product ,2. List product code wise, 3. List product category wise. Option 3 will provide you an option to update quantity of the specific item.Option 4 provides you with delete option to remove any item from the stock.

1. If Purchase management is selected, you will get a list of options which is as follows :

1. Add Order

2. List Order

3. Back (Main Menu)

Option 1 will provide you with an option to add any new ordermade. Option 2 will show the list of the orders made before.

1. If sales management is selected, you will get a list of options which is as follows :

1. Sale Items

2. List Sales

3. Back (Main Menu)

Option 1 will provide with an option to add the item which has to be sold, then enter the quantity, the total price which has to be collected will be provided on the screen. Option 2 will give the list of the sales made.

1. If sales management is selected, you will get a list of options which is as follows :

1. Add user

2. List user

3. Back (Main Menu)

Option 1 will provide you with an option to add a new user with email address, name and password. Option 2 will show the list of all users pre-existing in it.

(Note – If you have logged in with your specific user id and password you can directly access the user management option but if you have began with a new start you have to enter the management password to access the user management option. )

1. If sales management is selected, you will be asked for the management password, if you enter it correctly you will get a list of options which is as follows :

1. Database creation

2. List Database

3. Back (Main Menu)

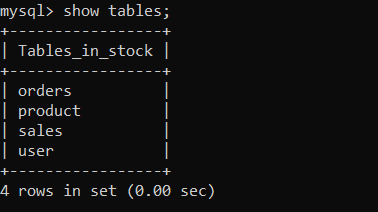
Option 1 is to create the setup and option 2 is to list the options created in the above database formed(data creation can also be done before form NEW START followed by CREATE A NEW DATABASE AND SWITCH TO MENU).

1. If exit option is selected the whole program will be closed.
2. If close pc option is selected, it will shut down your pc in next 1 second.

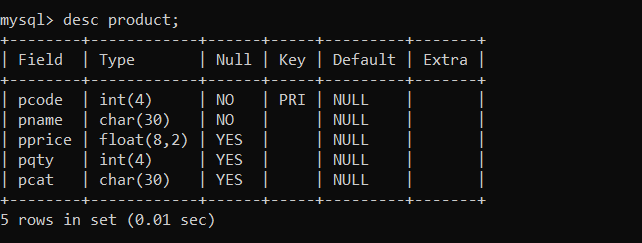
**LIBRARIES**

|  |  |  |
| --- | --- | --- |
| **Sr.no.** | **Name of libraries** | **Meaning** |
| 1. | os | The operating-system module contains a number of functions for manipulating information specific to a given application, rather than the environment as a whole. You can run or quit any application, and interrogate the running application for application-specific information. |
| 2. | Mysql connector | To access the MySQL database from Python, you need a database driver. MySQL Connector/Python is an API implemented using pure Python. |
| 3. | Datetime | The datetime module provides a number of types to deal with dates, times, and time intervals. ... The date type represents just a date, between year 1 and 9999 |

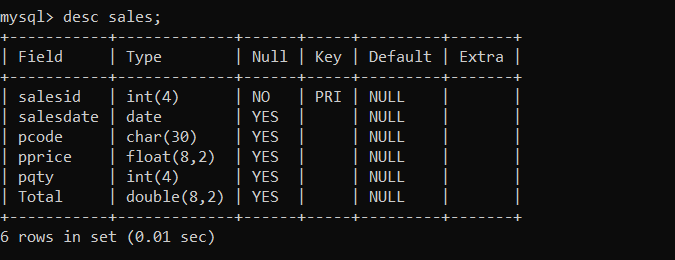
**MySql Tables**

****

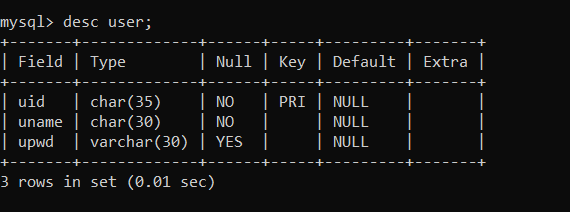
**Tables in stock database**

****

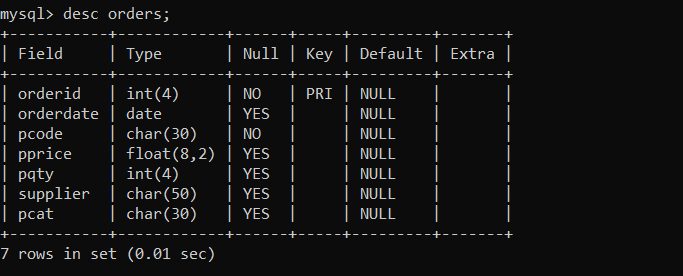
**Product table**

****

**Sales table**

****

**User table**

****

**Order table**

**SOURCE CODE**

# STOCK MANAGEMENT

import os

import mysql.connector

import datetime

import string

now = datetime.datetime.now()

def start():

print("\t\t\t STOCK MANAGEMENT")

print("\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

print("\t\t 1. NEW STOCK")

print("\t\t 2. OLD STOCK\n")

n=int(input("Enter your choice :"))

if n== 1:

print("\t 1. CREATE A NEW DATABASE ")

print("\t 2. OPEN MAIN MENU")

print("\t 3. BACK\n")

a=int(input("Enter your choice :"))

if a==1:

create\_database()

clrscr()

if a==2:

clrscr()

if a==3:

start()

if n== 2:

login2()

def login2():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345", database="stock") #as per system

mycursor=mydb.cursor()

a=input("Enter the user id.:")

b=len(a)

sql="SELECT uname FROM user where uname=%s"

val=(a,)

mycursor.execute(sql,val)

c=mycursor.fetchall()

f=str(c)

if a==(f[3:b+3]):

code=input("Enter the password:")

z=len(code)

abc="SELECT upwd FROM user where uname=%s"

val=(a,)

mycursor.execute(abc,val)

x=mycursor.fetchall()

w=str(x)

if code==(w[3:z+3]):

clrscr2()

else:

print("user name is invalid, plese try it again")

login2()

def clrscr2():

print("\n"\*5)

while True:

print("\t\t\t STOCK MANAGEMENT")

print("\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

print("\t\t 1. PRODUCT MANAGEMENT")

print("\t\t 2. PURCHASE MANAGEMENT")

print("\t\t 3. SALES MANAGEMENT")

print("\t\t 4. USER MANAGEMENT")

print("\t\t 5. DATABASE SETUP")

print("\t\t 6. EXIT")

print("\t\t 7. CLOSE PC\n")

n=int(input("Enter your choice :"))

if n== 1:

product\_mgmt()

if n== 2:

os.system('cls')

purchase\_mgmt()

if n==3:

sales\_mgmt()

if n== 4:

user\_mgmt( )

if n==5:

db\_mgmt()

if n== 6:

break

if n==7:

sht()

def clrscr():

print("\n"\*5)

while True:

print("\t\t\t STOCK MANAGEMENT")

print("\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

print("\t\t 1. PRODUCT MANAGEMENT")

print("\t\t 2. PURCHASE MANAGEMENT")

print("\t\t 3. SALES MANAGEMENT")

print("\t\t 4. USER MANAGEMENT")

print("\t\t 5. DATABASE SETUP")

print("\t\t 6. EXIT")

print("\t\t 7. CLOSE PC\n")

n=int(input("Enter your choice :"))

if n== 1:

product\_mgmt()

if n== 2:

os.system('cls')

purchase\_mgmt()

if n==3:

sales\_mgmt()

if n== 4:

deft()

if n==5:

db\_mgmt()

if n== 6:

break

if n==7:

sht()

def product\_mgmt( ):

while True :

print("\t\t\t 1. Add New Product")

print("\t\t\t 2. List Product")

print("\t\t\t 3. Update Product")

print("\t\t\t 4. Delete Product")

print("\t\t\t 5. Back (Main Menu)")

p=int (input("\t\tEnter Your Choice :"))

if p==1:

add\_product()

if p==2:

search\_product()

if p==3:

update\_product()

if p==4:

delete\_product()

if p== 5:

break

def purchase\_mgmt( ):

while True :

print("\t\t\t 1. Add Order")

print("\t\t\t 2. List Order")

print("\t\t\t 3. Back (Main Menu)")

o=int(input("\t\tEnter Your Choice :"))

if o==1 :

add\_order()

if o==2 :

list\_order()

if o== 3 :

break

def sales\_mgmt( ):

while True :

print("\t\t\t 1. Sale Items")

print("\t\t\t 2. List Sales")

print("\t\t\t 3. Back (Main Menu)")

s=int (input("\t\tEnter Your Choice :"))

if s== 1 :

sale\_product()

if s== 2 :

list\_sale()

if s== 3 :

break

def user\_mgmt( ):

while True :

print("\t\t\t 1. Add user")

print("\t\t\t 2. List user")

print("\t\t\t 3. Back (Main Menu)")

u=int (input("\t\tEnter Your Choice :"))

if u==1:

add\_user()

if u==2:

list\_user()

if u==3:

break

def deft():

a=input("Enter the management password:")

if a=='stock@management':

user\_mgmt()

else:

print("Password is incorrect, please try again")

deft()

def create\_database():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock") #change as per system

mycursor=mydb.cursor()

print(" Creating PRODUCT table")

sql = "CREATE TABLE if not exists product (\

pcode int(4) PRIMARY KEY,\

pname char(30) NOT NULL,\

pprice float(8,2) ,\

pqty int(4) ,\

pcat char(30));"

mycursor.execute(sql)

print("PRODUCT table created")

print(" Creating ORDER table")

sql = "CREATE TABLE if not exists orders (\

orderid int(4)PRIMARY KEY ,\

orderdate DATE ,\

pcode char(30) NOT NULL , \

pprice float(8,2) ,\

pqty int(4) ,\

supplier char(50),\

pcat char(30));"

mycursor.execute(sql)

print(" ORDER table created")

print(" Creating SALES table")

sql = "CREATE TABLE if not exists sales (\

salesid int(4) PRIMARY KEY ,\

salesdate DATE ,\

pcode char(30) references product(pcode), \

pprice float(8,2) ,\

pqty int(4) ,\

Total double(8,2)\

);"

mycursor.execute(sql)

print(" SALES table created")

print(" Creating USER table")

sql = "CREATE TABLE if not exists user (\

uid char(20) PRIMARY KEY,\

uname char(30) NOT NULL,\

upwd varchar(30));"

mycursor.execute(sql)

print(" USER table created")

def list\_database():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock") #as per system

mycursor=mydb.cursor()

sql="show tables;"

mycursor.execute(sql)

for i in mycursor:

print(i)

def add\_order():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="",database="stock") #as per system

mycursor=mydb.cursor()

now = datetime.datetime.now()

sql="INSERT INTO orders (orderid, orderdate, pcode, pprice, pqty, supplier, pcat) values (%s,%s,%s,%s,%s,%s,%s)"

code=int(input("Enter product code :"))

oid=now.year+now.month+now.day+now.hour+now.minute+now.second

qty=int(input("Enter product quantity : "))

price=float(input("Enter Product unit price: "))

cat=input("Enter product category: ")

supplier=input("Enter Supplier details: ")

val=(oid,now,code,price,qty,supplier,cat)

mycursor.execute(sql,val)

mydb.commit()

def list\_order():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345", database="stock") #as per system

mycursor=mydb.cursor()

sql="SELECT \* from orders"

mycursor.execute(sql)

print("\t\t\t\t\t\t\t ORDER DETAILS")

print("-"\*120)

print("orderid Date Product code price quantity Supplier Category")

print("-"\*120)

for i in mycursor:

print(i[0],"\t",i[1],"\t",i[2],"\t ",i[3],"\t",i[4],"\t ",i[5],"\t",i[6])

print("-"\*120)

def db\_mgmt( ):

while True :

print("\t\t\t 1. Database creation")

print("\t\t\t 2. List Database")

print("\t\t\t 3. Back (Main Menu)")

p=int (input("\t\tEnter Your Choice :"))

if p==1 :

create\_database()

if p==2 :

list\_database()

if p== 3 :

break

def add\_product():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock") #as per system

mycursor=mydb.cursor()

sql="INSERT INTO product(pcode,pname,pprice,pqty,pcat) values (%s,%s,%s,%s,%s)"

code=int(input("\t\tEnter product code :"))

search="SELECT count(\*) FROM product WHERE pcode=%s;"

val=(code,)

mycursor.execute(search,val)

for x in mycursor:

cnt=x[0]

cnt==0

name=input("\t\tEnter product name :")

qty=int(input("\t\tEnter product quantity :"))

price=float(input("\t\tEnter product unit price :"))

cat=input("\t\tEnter Product category :")

val=(code,name,price,qty,cat)

mycursor.execute(sql,val)

mydb.commit()

else:

print("\t\t Product already exist")

def update\_product():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock") #as per system

mycursor=mydb.cursor()

code=int(input("Enter the product code :"))

qty=int(input("Enter the quantity :"))

sql="UPDATE product SET pqty=pqty+%s WHERE pcode=%s;"

val=(qty,code)

mycursor.execute(sql,val)

mydb.commit()

print("\t\t Product details updated")

def delete\_product():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock") #as per system

mycursor=mydb.cursor()

code=int(input("Enter the product code :"))

sql="DELETE FROM product WHERE pcode = %s;"

val=(code,)

mycursor.execute(sql,val)

mydb.commit()

print(mycursor.rowcount," record(s) deleted");

def search\_product():

while True :

print("\t\t\t 1. List all product")

print("\t\t\t 2. List product code wise")

print("\t\t\t 3. List product categoty wise")

print("\t\t\t 4. Back (Main Menu)")

s=int(input("\t\tEnter Your Choice :"))

if s==1 :

list\_product()

if s==2 :

code=int(input(" Enter product code :"))

list\_prcode(code)

if s==3 :

cat=input("Enter category :")

list\_prcat(cat)

if s==4 :

break

def list\_product():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock") #as per system

mycursor=mydb.cursor()

sql="SELECT \* from product"

mycursor.execute(sql)

print("\t\t\t\t PRODUCT DETAILS")

print("\t\t","-"\*95)

print("\t\t code name price quantity category")

print("\t\t","-"\*95)

for i in mycursor:

print("\t\t",i[0],"\t",i[1],"\t",i[2],"\t",i[3],"\t\t",i[4])

print("\t\t","-"\*95)

def list\_prcode(code):

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock")

mycursor=mydb.cursor()

sql="SELECT \* from product WHERE pcode=%s"

val=(code,)

mycursor.execute(sql,val)

print("\t\t\t\t PRODUCT DETAILS")

print("\t\t","-"\*47)

print("\t\t code name price quantity category")

print("\t\t","-"\*47)

for i in mycursor:

print("\t\t",i[0],"\t",i[1],"\t",i[2],"\t ",i[3],"\t\t",i[4])

print("\t\t","-"\*47)

def sale\_product():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock") #as per system

mycursor=mydb.cursor()

pcode=input("Enter product code: ")

sql="SELECT count(\*) from product WHERE pcode=%s;"

val=(pcode,)

mycursor.execute(sql,val)

for x in mycursor:

cnt=x[0]

if cnt !=0 :

sql="SELECT \* from product WHERE pcode=%s;"

val=(pcode,)

mycursor.execute(sql,val)

for x in mycursor:

print(x)

price=int(x[2])

pqty=int(x[3])

qty=int(input("Enter no of quantity :"))

if qty <= pqty:

total=qty\*price;

print ("Collect Rs. ", total)

sql="INSERT into sales values(%s,%s,%s,%s,%s,%s)"

val=(int(cnt)+1,datetime.datetime.now(),pcode,price,qty,total)

mycursor.execute(sql,val)

sql="UPDATE product SET pqty=pqty-%s WHERE pcode=%s"

val=(qty,pcode)

mycursor.execute(sql,val)

mydb.commit()

else:

print(" Quantity not Available")

else:

print(" Product is not avalaible")

def list\_sale():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock") #as per system

mycursor=mydb.cursor()

sql="SELECT \* FROM sales"

mycursor.execute(sql)

print(" \t\t\t\tSALES DETAILS")

print("-"\*80)

print("Sales id Date Product Code Price Quantity Total")

print("-"\*80)

for x in mycursor:

print(x[0],"\t",x[1],"\t",x[2],"\t ",x[3],"\t\t",x[4],"\t\t",x[5])

print("-"\*80)

def list\_prcat(cat):

mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="stock") #as per system

mycursor=mydb.cursor()

print (cat)

sql="SELECT \* from product WHERE pcat =%s"

val=(cat,)

mycursor.execute(sql,val)

clrscr()

print("\t\t\t\t PRODUCT DETAILS")

print("\t\t","-"\*97)

print("\t\t code name price quantity category")

print("\t\t","-"\*97)

for i in mycursor:

print("\t\t",i[0],"\t",i[1],"\t",i[2],"\t ",i[3],"\t\t",i[4])

print("\t\t","-"\*97)

def add\_user():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="",database="stock") #as per system

mycursor=mydb.cursor()

uid=input("Enter emaid id :")

name=input(" Enter Name :")

paswd=input("Enter Password :")

sql="INSERT INTO user values (%s,%s,%s);"

val=(uid,name,paswd)

mycursor.execute(sql,val)

mydb.commit()

print(mycursor.rowcount, " user created")

def list\_user():

mydb=mysql.connector.connect(host="localhost",user="root",passwd="",database="stock") #as per system

mycursor=mydb.cursor()

sql="SELECT uid,uname from user"

mycursor.execute(sql)

print("\t\t\t\t USER DETAILS")

print("\t\t","-"\*67)

print("\t\t UID name ")

print("\t\t","-"\*67)

for i in mycursor:

print("\t\t",i[0],"\t",i[1])

print("\t\t","-"\*67)

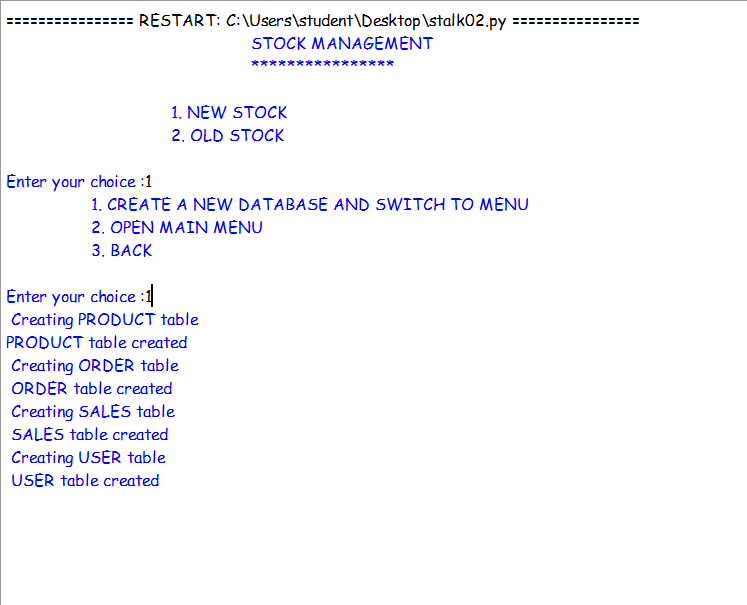
def sht():

os.system("shutdown /s /t 1")

start()

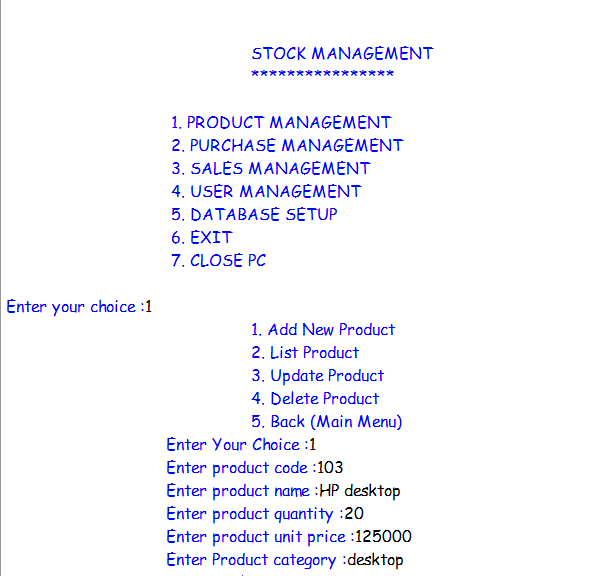
INPUT/OUTPUT

INTERFACE

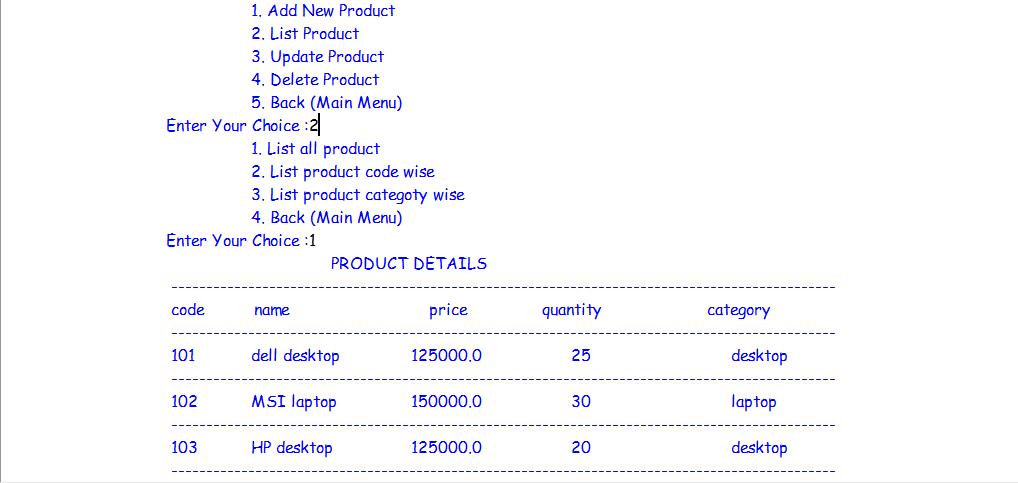
**First screen with a new start up.**

****

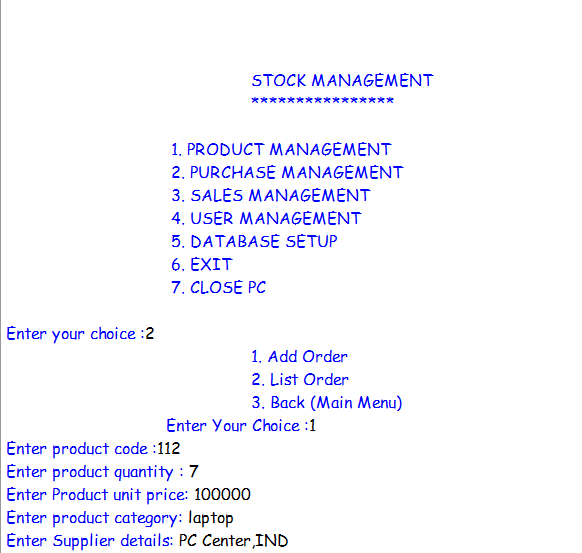
**First screen with old stock option.**

****

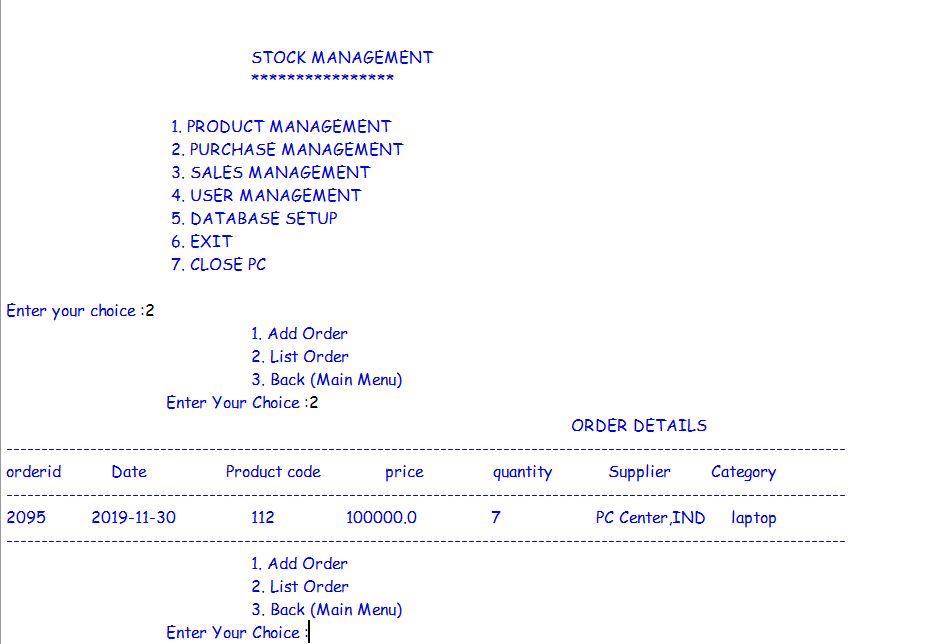
**Adding a new product.**

****

**Displaying list of products.**

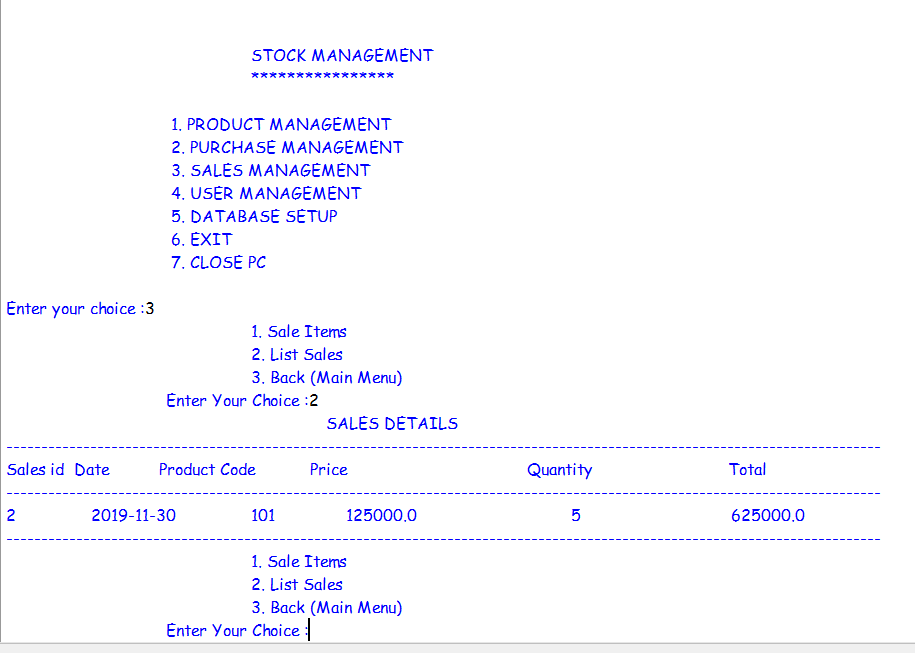
****

**Adding order to purchase management option.**

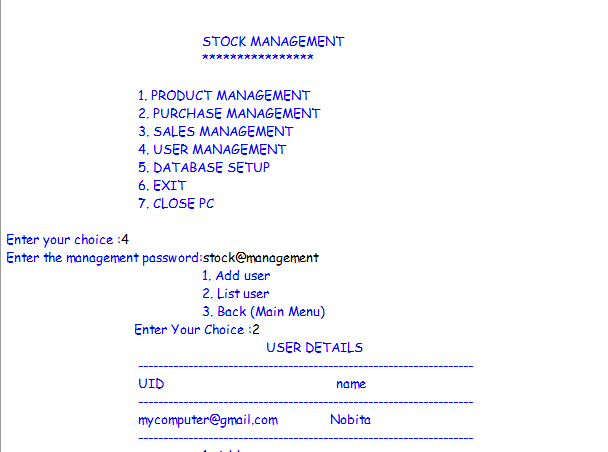
**Displaying list of orders made.**

**Selling item in sale management option**

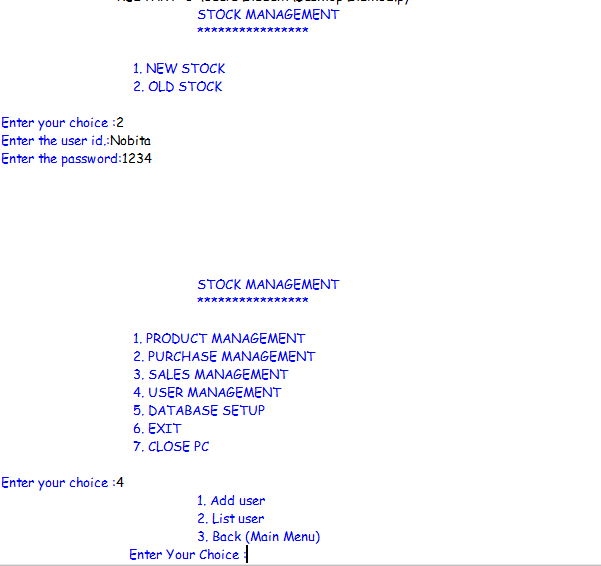
**Total amount to be paid is displayed.**

****

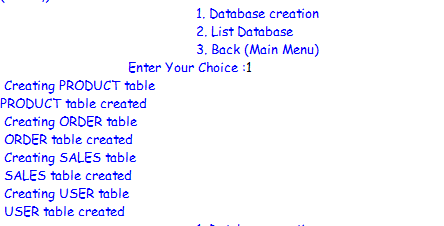
**Displaying the list of sales made.**

****

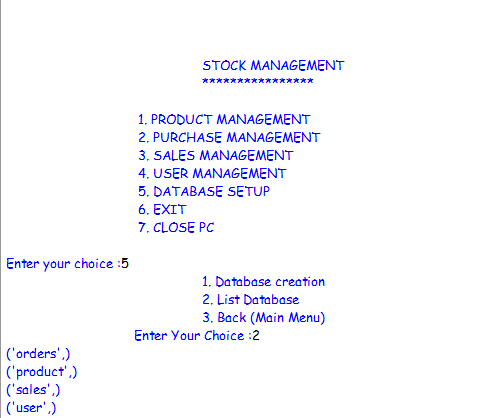
* **Operating user management option with new stock mode, as the password is asked.**
* **List of user is also displayed.**

****

**Operating user management option with old stock mode, as management password is not asked.**

****

**Creating tables in database.**

****

**Displaying list of tables in database.**

**Operating exit option.**

**TECHNOLOGICAL SPECIFICATIONS:**

**SOFTWARE SPECIFICATION:-**

Operating System : Windows 10

Platform : Python IDLE 3.7

Database : MySQL

Languages : Python

**HARDWARE SPECIFICATION:-**

Processor : i7 Dual Core

Hard Disk : 02TB

Ram : 8024 MB

**Note:** For Python-MySQL connectivity, following data have been used:-

**Host- localhost, user- root, password- 12345, database- stock**

**Management password-**stock@management

**BIBLIOGRAPHY:**

* Computer science with python Sumita Arora class 12
* Computer science with python Sumita Arora class 11
* <https://www.learnpython.org/>
* <https://docs.python-guide.org/intro/learning/>