****

**OOP PROJET REPORT**

**PHARMACY MANAGEMENT SYSTEM**

**SUBMITTED TO : SIR NADEEM YOUSAF**

**SUBMITTED BY :**

**ABDUL RAHIM - 210201097**

**SYED DAWOOD SHAH - 210201004**

**MUHAMMAD HUZAIFA NASIR - 210201047**

**RAZA ULLAH - 210201099**

**SAAD ASHRAF – 210201064**

1. Project Name :

Pharmacy Management System

1. Introduction :

In this Project, we are going to design a for a clothing store. In which we can add Products as in a database and we can work on them like making a purchase, Updating Id, deleting them or simple a Proper Receipt printer; etc.

1. Methodology :

In this code, the username, password will be asked at first but before that the user will be asked that how many products the user want to add. After the credentials are verified the product details will be added in the database and after that simply we can perform different functions on that app just like:

Show all Products

* Add new Products
* Search By ID’s
* Delete ID’s
* Update ID’s
* Make a Purchase / Single – Multiple

Working :

**User Side:**

* User asked to enter the Username, Password.
* User asked to add Product Details.
* User Will get Multiple options to perform Different operations.
* In one Option i.e., User can make a purchase by providing id & Stock.

**Developer Side:**

* Declared variables that are going to be used in Program (string, int variables for input and results).
* According to the demand the desired operations can be Performed, Options are available in POINT - 3
* According to the Total Price the Discounts will be Given Up to been Displayed.

1. Concepts Used :
   * Classes
   * Abstract Class
   * Static Method
   * Objects
   * Encapsulation
   * Exceptional Handling
   * Polymorphism
   * Setter & Getter
   * Information Hiding
   * Lists
   * Functions
   * Different loops(if, else if, While, For)
   * Random, ABC, sys Library Imported
2. Code :

from random import randint

from abc import ABC, abstractmethod

import sys

r = randint(10000,99999)

print("\nWelcome To Pharmacy Management System!\n") # Basic name display!

class Error(Exception): # class made for exceptional handling to be used later!

    pass

class ValueTooSmallError(Error): # class made for exceptional handling to be used later!

    pass

class abstract(ABC):

    \_x = False # to be used in credentials function in next class as a checker in check function to determine weather

                #the program will enter the loop outside the classes or not!

    @abstractmethod

    def credentials(): #used as a blueprint! Will be used later as to be saied that it will be overwritten (that's the purpose).

        pass

    def checking():

        pass

class Medicine:

    \_\_y = False

    admin = 'pharmacy'

    pswd = 'qqq'

    @property

    def Get\_Details(self): #getting 4 things from the user: (Id, Rate, Power & Stock)

        self.\_\_id = input("\nEnter Medicine ID: ")

        while True:

            try:

                self.rate = int(input("Enter Rate Per Packet: "))

                if (self.rate <= 0):

                  raise ValueTooSmallError

                break

            except ValueError as e:

                print("\nINVALID LITERAL for int() with Base 10", e)

                print("Try again!")

            except ValueTooSmallError:

               print("Price can Never be Zero-0 or any Negative Value!")

        while True:

            try:

                self.power = int(input("Power of that Medicne/Tablet (i.e; 250mg, 500mg): "))

                if (self.power <= 0):

                  raise ValueTooSmallError

                break

            except ValueError as e:

                print("\nINVALID LITERAL for int() with Base 10", e)

                print("Try again!")

            except ValueTooSmallError:

               print("Price can Never be Zero-0 or any Negative Value!")

        while True:

            try:

                self.stock = int(input("Enter Available Stock: "))

                if (self.stock <= 0):

                    raise ValueTooSmallError

                break

            except ValueError as e:

                print("\nINVALID LITERAL for int() with Base 10", e)

                print("Try again!")

            except ValueTooSmallError:

               print("Stock can only be in integer & also not any Negative value\nTry again!")

    def credentials(abstract): #person is being called here from abstract class to be overwritten!

        for i in range(3): # for username

            m = input("\nLogin ID: ").lower()

            if (m == Medicine.admin):

                Medicine.\_\_y = True

                break

            else:

                print("\nWrong Details!")

                abstract.\_x = False

        if (Medicine.\_\_y == True):

            for j in range(3): #for password

                w = input("Enter Passcode: ").lower()

                if (w == Medicine.pswd):

                    P.Get\_Details

                    print("\nYOU HAVE BEEN SUCCESSFULLY LOGGED IN!")

                    abstract.\_x = True

                    break

                else:

                    print("\nWrong Passcode!")

                    abstract.\_x = False

        else:

            abstract.\_x = False

    def checking(abstract):

        if abstract.\_x == True:

            return True

        else:

            print("\nYou have Entered Wrong Details!\nYou have been LOGGED OUT!")

            return False

    def Display\_Details(self):

        print("(ID:",self.\_\_id, ") (Price:",self.rate, ") (Power:",self.power, ") (Stock:",self.stock,")")

    def Setter\_id(self, id): #being used her as a searcher to verify that if indeed something exists or not!

      if self.\_\_id == id:   #also Setter from Setter - Getter!

          return True

      else:

          return False

    def getter\_id(self):  #Simply Getter from Setter - Getter!

      return self.\_\_id

    def Sale(self):    # purchasing

        print("\nPresent Stock is:", self.stock)

        d = input("Enter Purchaser Name: ")

        if(self.stock > 0):

            q = int(input("Enter Quantity to Buy: "))

            if (q <= self.stock):

                total = q \* self.rate

                print("\nAmount:",total)

                self.stock -= q

                disc = 0

                if total in range(0,500):

                    disc = 0 / 100 \* total

                    finalp = total - disc

                    print("\nDiscount - 0% :",disc)

                elif total in range(501,1000):

                    disc = 5 / 100 \* total

                    finalp = total - disc

                    print("\nDiscount - 5% :",disc)

                elif total in range(1001,1500):

                    disc = 10 / 100 \* total

                    finalp = total - disc

                    print("\nDiscount - 10% :",disc)

                elif total in range(1501,2500):

                    disc = 20 / 100 \* total

                    finalp = total - disc

                    print("\nDiscount - 20% :",disc)

                else:

                    disc = 25 / 100 \* total

                    finalp = total - disc

                    print("\nDiscount - 25% :",disc)

                #Receipt:

                print("\n-----------------------")

                print("-----------------------")

                print("----------PMS----------")

                print("-----------------------")

                print(" "\*18 ,r)

                print("Your Medicine is:\n(",self.\_\_id,")")

                print('\nDeduction = ',total,"-",disc)

                print("BILL:",finalp,"Rs only!\n")

                print("\nThank- You!", d)

            else:

                print("Not Enough Quantity Available!")

        else:

            print("Not Enough Stock Available!")

    def display(self,id):   # to simply change name of any Medicine

        self.\_\_id = input("Enter new ID: ")

        print("\nUpdated Name is:", self.\_\_id)

    def updater\_stock(self,id):

        q = input("Add or New: ").lower()

        if q == 'add':

            n = int(input("Add/Subtract Stock: "))

            self.stock += n

            print("\nUpdated Stock of",self.\_\_id, "is: ", self.stock)

        else:

            while True:

                try:

                    self.stock = int(input("New Stock: "))

                    if (self.stock <= 0):

                        raise ValueTooSmallError

                    print("\nUpdated Stock of",self.\_\_id, "is: ", self.stock)

                    break

                except ValueError as e:

                    print("\nINVALID LITERAL for int() with Base 10", e)

                    print("Try again!")

                except ValueTooSmallError:

                    print("Stock can only be in integer & also not any Negative value\nTry again!")

class End(Medicine):

    @staticmethod

    def display():

        print("\nThank-You!\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nProject By: Abdul Rahim, Syed Dawood, Raza Ullah, Huzaifa Nasir & Saad Ashraf\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

# ----------------------

L = [] # to append all details!

S = End()

try:

    n = int(input("How many Medical-Items to add: "))

except ValueError:

    print("Invalid Literal for int() with Base - 10: ")

for i in range(n):

    P = Medicine()

    P.credentials()

    L.append(P)

# ----------------------

while P.checking():

    print("""\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    Main - Menu:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*""")

    print("\n1-  Show All Medics Details.\n2-  Add New Information.\n3-  Search By ID/Name.\n4-  Delete Single Item.\n5-  Delete All Records.\n6-  Purchase.\n7-  Purchase Multiple.\n8-  Update ID.\n9-  Update Stock.\n10- Exit\n")

    ch = int(input("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nEnter Your Choice: "))

    if ch == 1:

        print("\nMedics Available:\n")

        for c in L:

            if L == []:

                print("No Medics Available in Stock!")

            else:

                c.Display\_Details()

    elif ch == 2:

        o = int(input("\nHow Many More items to Add: "))

        for i in range(o):

            P = Medicine()

            P.credentials()

            L.append(P)

    elif ch == 3:

        id = input("\nEnter Medic ID/Name to Search: ")

        found = False

        for c in L:

            found = c.Setter\_id(id)

            if found == True:

                print("Details From ID Found:")

                c.Display\_Details()

                break

        if not found:

            print("Not - Found!")

    elif ch == 4:

        id = input("\nEnter Medicine ID to Delete: ")

        found = False

        for c in L:

            found = c.Setter\_id(id)

            if found == True:

                L.remove(c)

                print("Medic-Item has Been Deleted!")

                break

        if not found:

            print("Not - Found!")

    elif ch == 5:

        L.clear()

        print("All Medical Details has been Removed!\nDatabase is EMPTY!")

    elif ch == 6:

        q = input("\nEnter ID/Name of Medical Item to Purchase: ")

        found = False

        for c in L:

            found = c.Setter\_id(q)

            if found == True:

                print("\nRECEIPT:")

                c.Sale()

                S.display()

        if not found:

            print("Not - Found!")

    elif ch == 7:

        i = int(input("How Many Medical items to Purchase: "))

        for r in range(i):

            q = input("\nEnter ID/Name of Medical Item to Purchase: ")

            found = False

            for c in L:

                found = c.Setter\_id(q)

                if found == True:

                    print("\nRECEIPT:")

                    c.Sale()

                    S.display()

            if not found:

                print("Not - Found!")

    elif ch == 8:

        id = input("\nEnter Medic ID/Name to Update: ")

        found = False

        for c in L:

            found = c.Setter\_id(id)

            if found == True:

                print("ID Found:")

                c.display(id)

                break

        if not found:

            print("Not - Found")

    elif ch == 9:

        id = input("\nEnter Product ID to Update Stock in: ")

        found = False

        for c in L:

            found = c.Setter\_id(id)

            if found == True:

                print("ID Found:")

                c.Display\_Details()

                c.updater\_stock(id)

                break

        if not found:

            print("Not - Found")

    elif ch == 10:

        S.display()

        break

    else:

        print("Invalid Choice - Try Again!")

6.Result :

Our Pharmacy Name will be Displayed, and the Username - Password will be asked but before that Appending data total info will be required which is stored in a List!

If perhaps the user enter wrong credentials the user will have 3 chances to have it right otherwise the program will end!

Welcome To Pharmacy Management System!

How many Medical-Items to add: 1

Login ID: pharmacy

Enter Passcode: qqq

# Then, the administrator will add details to add in the database. After that user will be logged in & and a Menu will be displayed to choose options to perform different operations on Further!

Enter Medicine ID: panadol

Enter Rate Per Packet: 20

Power of that Medicne/Tablet (i.e; 250mg, 500mg): 250

Enter Available Stock: 50

YOU HAVE BEEN SUCCESSFULLY LOGGED IN!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Main - Menu:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1- Show All Medics Details.

2- Add New Information.

3- Search By ID/Name.

4- Delete Single Item.

5- Delete All Records.

6- Purchase.

7- Purchase Multiple.

8- Update ID.

9- Update Stock.

10- Exit

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter Your Choice:

Simply, In Option – 1, All available details Will be displayed!

And after every successful operation, Menu will open again!

Medics Available:

(ID: panadol ) (Price: 20 ) (Power: 250 ) (Stock: 50 )

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Main - Menu:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1- Show All Medics Details.

2- Add New Information.

3- Search By ID/Name.

4- Delete Single Item.

5- Delete All Records.

6- Purchase.

7- Purchase Multiple.

8- Update ID.

9- Update Stock.

10- Exit

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter Your Choice:

In Option – 2, We can add new Products (Details…)

* Indeed, Credentials will be asked again as It’s the management system Privacy!
* And as were told before exceptional – Handling was used to get correct details.

So, in Option – 2 , user will be asked that how many the user would like to add more, then again credentials and then the details will be added!

Enter Your Choice: 2

How Many More items to Add: 1

Login ID: pharmacy

Enter Passcode: qqq

Enter Medicine ID: 2

Enter Rate Per Packet: 2

Power of that Medicne/Tablet (i.e; 250mg, 500mg): 20

Enter Available Stock: 20

In Option – 3, An Id has to be given to search for its details simply!

Enter Your Choice: 3

Enter Medic ID/Name to Search: 1

Details From ID Found:

(ID: 1 ) (Price: 1 ) (Power: 1 ) (Stock: 1 )

# In Option – 4, An Id has to be Given and all its details will be Erased from Existence!

Enter Your Choice: 4

Enter Medicine ID to Delete: 1

Medic-Item has Been Deleted!

# In Option – 5, All Product Details will be Erased from Existence!

Enter Your Choice: 5

All Medical Details has been Removed!

Database is EMPTY!

# In Option – 6, We can make a Purchase, Id has to be given and amount of stock we would like to purchase, of course E-H is still used here, then a customer name and a receipt will be printed!

Enter ID/Name of Medical Item to Purchase: 1

RECEIPT:

Present Stock is: 1

Enter Purchaser Name: Rahim

Enter Quantity to Buy: 1

Amount: 1

Discount - 0% : 0.0

-----------------------

-----------------------

----------PMS----------

-----------------------

80780

Your Medicine is:

( 1 )

Deduction = 1 - 0.0

BILL: 1.0 Rs only!

Thank- You! Rahim

Thank-You!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Project By: Abdul Rahim, Syed Dawood, Raza Ullah, Huzaifa Nasir & Saad Ashraf

In Option – 7, The same will be done but as many time as we want with a loop and a different Receipt for different Purchases!

In Option – 8, An Id has to be given to update its Name!

Enter Your Choice: 8

Enter Medic ID/Name to Update: 1

ID Found:

Enter new ID: 2

In Option 9 – , To update stock.

Enter Your Choice: 9

Enter Product ID to Update Stock in: 2

ID Found:

(ID: 2 ) (Price: 1 ) (Power: 1 ) (Stock: 0 )

Add or New: 1

New Stock: 2

Updated Stock of 2 is: 2

# In Option – 10, We will simply exit from the System! Indeed with Ending Details!

Enter Your Choice: 10

Thank-You!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Project By: Abdul Rahim, Syed Dawood, Raza Ullah, Huzaifa Nasir & Saad Ashraf

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**THE END**