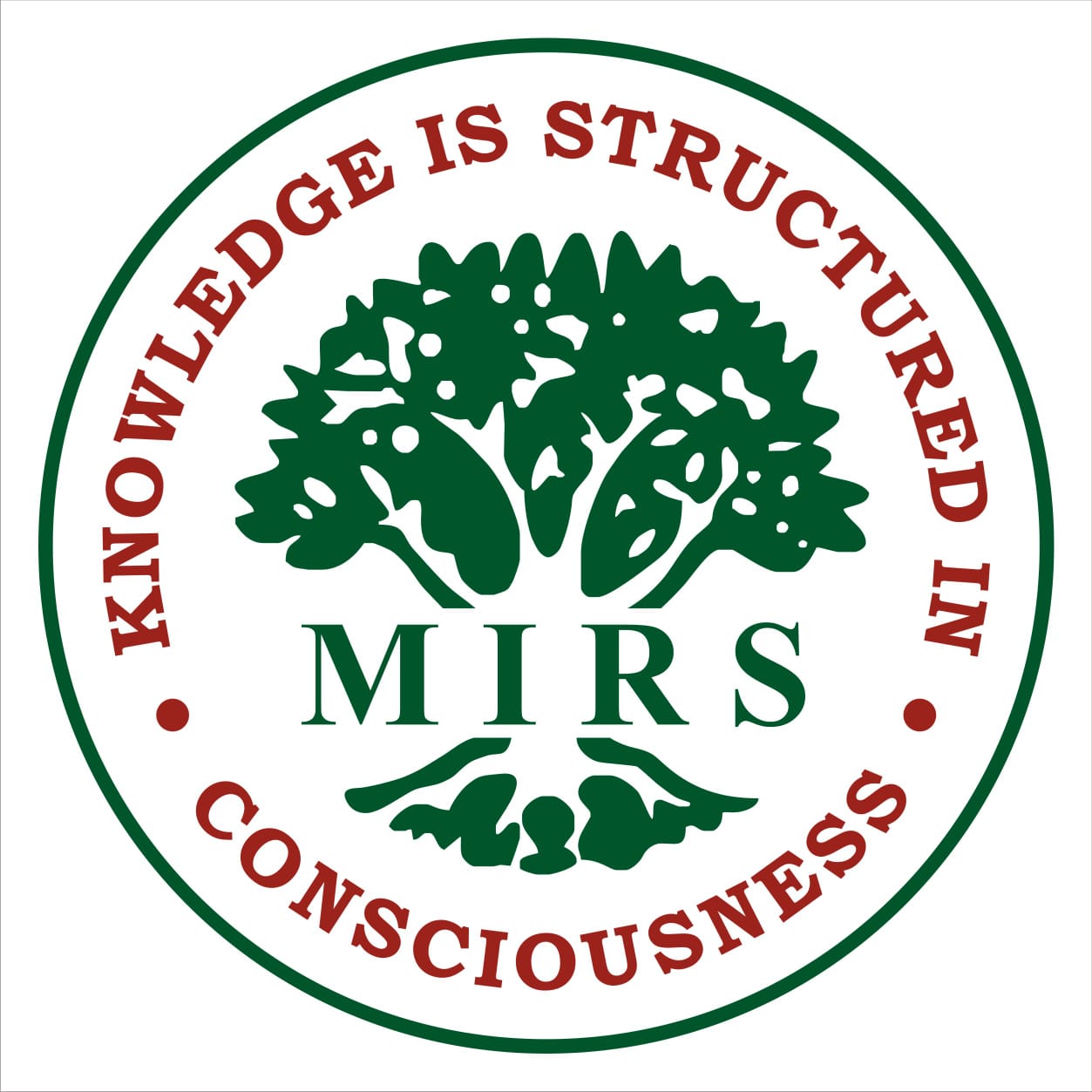
MAHARISHI INTERNATIONAL RESIDENTIAL SCHOOL



**COMPUTER SCIENCE (083)**

**PROJECT**

**2022-2023**

**TOPIC:**

**MEDICINE STOCK CHECKING** **SYSTEM**

**SUBMITTED BY : %name%**

**CLASS AND SECTION : %class%-%section%**

**TABLE OF CONTENTS**

* **INTRODUCTION TO PYTHON.**
* **INTRODUCTION TO THE PROJECT.**
* **SYSTEM REQUIREMENTS.**
* **BACKEND DETAILS.**
* **FRONTEND DETAILS.**
* **MOTIVE.**
* **SCREEN SHOTS OF EXECUTION.**
* **BIBLIOGRAPHY.**
* **LIMITATIONS.**

**INTRODUCTION TO PYTHON**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**History of Python:**

Python is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.



**INTRODUCTION TO THE PROJECT**

The Medicine Stock Checking System software is an ERP software used by medicine shops or medicine dealers for wholesale/retail business. This software stores details of medicines and helps us to search medicines by their name and manufacturer. It is possible to edit medicine cost and sell the medicine. The balance i.e. due amount of the stock can also be checked. If the medicine is expired ,the system has the provision to dispose it to the system specified warehourse. The program is also useful to check the details of the expired medicines.

**SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENT:**

* Printer- to print the required documents of the project.
* Compact Drive
* Proccesor: Pentium III and above
* RAM: 256 MB(minimum)
* Hard-Disk : 20 GB(minimum)

**SOFTWARE REQUIREMENT:**

* Windows 7 or higher
* My-SQL server 5.5 or higher (as backend)
* Python idle 3.6 or higher or spyder (as frontend).
* Microsoft Word 2010 or higher for documentation.

**BACKEND DETAILS**

**Database Name: MEDICINE**

**Code:**

Create Database Medicine;

Use Medicine;

**Table Name: STOCK**

**Attributes:**

Batch\_no int(11) Primary Key

name varchar(50)

manuf varchar(50)

date\_man date

date\_exp date

quantity int

sell int

balance int

cost\_unit int

**Code:**

CREATE TABLE STOCK (

Batch\_no int(11) Primary Key,

name varchar(50),

manuf varchar(50) ,

date\_man date,

date\_exp date,

quantity int,

sell int,

balance int,

cost\_unit int );

**Table Name: DISPOSE**

**Attributes:**

Batch\_no int(11)

Name varchar(50)

date\_exp date

amount int

**Code:**

CREATE TABLE DISPOSE (

Batch\_no int(11),

Name varchar(50),

date\_exp date,

amount int );

**FRONT-END DETAILS**

**PROGRAM CODE**

import sys

import random

import datetime

import mysql.connector

import matplotlib.pyplot as plt

mycon=mysql.connector.connect(host='localhost',user='root',password='abhisek',database='medicine')

mycur=mycon.cursor()

def Store():

sql="Insert into stock(batch\_no,name,manuf,date\_man,date\_exp,quantity,sell,balance,cost\_unit)values(%s,%s,%s,%s,%s,%s,%s,%s,%s)"

print('\nPLEASE PROVIDE THE REQUIRED INFORMATION\n')

acc=int(input('\nENTER THE BATCH NUMBER:'))

nm=input('\nENTER THE NAME OF THE MEDICINE WITH POWER:')

addr=input('\nENTER THE NAME OF THE MANUFACTURER:')

dbs=input('\nENTER THE DATE OF MANUFACTURE(YYYY-MM-DD):')

dacc=input('\nENTER THE DATE OF EXPIRY(YYYY-MM-DD):')

quan=int(input('\nENTER THE QUANTITY OF THE IMPORTED MEDICINE:'))

sell=0

balance=quan

cost=int(input('\nENTER THE COST OF THE IMPORTED MEDICINE PER UNIT:'))

value=(acc,nm,addr,dbs,dacc,quan,sell,balance,cost)

try:

mycur.execute(sql,value)

print(nm,'ADDED TO THE STOCK')

mycon.commit()

except:

print('UNABLE TO ADD MEDICINE!!!!!')

def Search\_by\_Name():

ph=input('\nENTER THE MEDICINE NAME TO SEARCH:')

sql="Select \* from Stock where name=%s"

value=(ph,)

mycur.execute(sql,value)

rec=mycur.fetchone()

if(rec==None):

print(ph,'IS NOT AVAILABLE')

else:

print('BATCH NUMBER:\t',rec[0])

print('MEDICINE NAME:\t',rec[1])

print('MANUFACTURER:\t',rec[2])

print('DATE OF MANUFACTURE:\t',rec[3])

print('DATE OF EXPIRY:\t',rec[4])

print('QUANITTY STORED:\t',rec[5])

print('INITIAL COST:\t',rec[8])

def Search\_by\_Manu():

ph=input('\nENTER THE MANUFACTURER NAME TO SEARCH:')

sql="Select name from Stock where manuf=%s"

value=(ph,)

mycur.execute(sql,value)

rec=mycur.fetchall()

if(rec==None):

print(ph,'IS A WRONG MANUFACTURER')

else:

print('----------MEDICINES MANUFACTURED BY',ph,'--------------------')

for nm in rec:

print(nm[0])

def Cost\_Update():

sql="Update stock set cost\_unit=%s where name=%s";

ph=input('\nENTER THE MEDICINE NAME TO CHANGE COST:')

addr=int(input('\nENTER THE NEW COST PER UNIT:'))

value=(addr,ph)

try:

mycur.execute(sql,value)

mycon.commit()

print('NEW COST OF',ph,'IS=RS',addr)

except:

print('UNABLE TO CHANGE COST!!!!')

def Sell():

sql="Update stock set sell=%s,balance=%s where name=%s";

ph=input('\nENTER THE MEDICINE NAME TO SELL:')

addr=int(input('\nENTER THE QUANTITY TO SELL:'))

sql2='select quantity from stock where name=%s'

value2=(ph,)

mycur.execute(sql2,value2)

rec=mycur.fetchone()

if(addr>rec[0]):

print('INSUFFICIENT STOCK IN HAND!!!!!!')

return

else:

balance=rec[0]-addr

value=(addr,balance,ph)

try:

mycur.execute(sql,value)

mycon.commit()

print(addr,'UNITS OF',ph,'SOLD')

print(balance,'UNITS LEFT')

except:

print('UNABLE TO SELL MEDICINE!!!!')

def Available():

ph=input('\nENTER THE MEDICINE NAME TO SEARCH:')

sql="Select balance from Stock where name=%s"

value=(ph,)

mycur.execute(sql,value)

rec=mycur.fetchone()

if(rec==None):

print(ph,'IS NOT AVAILABLE')

else:

print(rec[0],'UNITS OF',ph,'IS AVAILABLE')

def Dispose():

sql="Insert into dispose(batch\_no,name,date\_exp,amount)values(%s,%s,%s,%s)"

nm=input('\nENTER THE MEDICINE NAME TO DISPOSE:')

sql2="Select batch\_no,name,date\_exp,balance from stock where name=%s and date\_exp<=%s"

t\_date=datetime.date.today()

value2=(nm,t\_date)

mycur.execute(sql2,value2)

rec=mycur.fetchone()

if(rec==None):

print(nm,'IS NOT EXPIRED YET')

else:

print(nm,'IS EXPIRED')

c=int(input('\nPRESS 1 TO DISPOSE IT:'))

if(c==1):

b=rec[0]

n=rec[1]

d=rec[2]

am=rec[3]

value=(b,n,d,am)

sql3='Delete from stock where name=%s'

value3=(n,)

try:

mycur.execute(sql,value)

mycon.commit()

print(n,'SUCCESSFULLY DISPOSED')

mycur.execute(sql3,value3)

mycon.commit()

except:

print('UNABLE TO DISPOSE MEDICINE')

else:

print('WARNING!!!!!',nm,'MUST BE DISPOSED LATER')

return

def Search\_Dispose():

ph=input('\nENTER THE DISPOSED MEDICINE NAME TO SEARCH:')

sql="Select \* from Dispose where name=%s"

value=(ph,)

mycur.execute(sql,value)

rec=mycur.fetchone()

if(rec==None):

print(ph,'IS NOT AVAILABLE')

else:

print('BATCH NUMBER:\t',rec[0])

print('MEDICINE NAME:\t',rec[1])

print('DATE OF EXPIRY:\t',rec[2])

print('BALANCE AMOUNT:\t',rec[3])

def Graph():

ad=input('ENTER MEDICINE NAME:')

sql='Select \* from stock where name=%s'

value=(ad,)

mycur.execute(sql,value)

T=mycur.fetchone()

N=[T[5],T[6],T[7]]

L=['QUANTITY','SELL','BALANCE']

clr=('red','blue','green')

plt.bar(L,N,color=clr)

plt.xlabel('MEDICINE STATUS')

plt.ylabel('VALUES')

plt.title('MEDICINE QUANITTY-SELL-BALANCE')

plt.show()

def Close():

print('\nTHANK YOU FOR USING THE APPLICATION')

sys.quit()

print('------------WELCOME TO MEDICINE STOCK CHECKING SYSTEM-------------\n\n')

while(True):

print('\n\nPRESS 1 TO ADD A NEW MEDICINE')

print('PRESS 2 TO SEARCH A MEDICINE BY NAME')

print('PRESS 3 TO SEARCH A MEDICINE BY MANUFACTURER')

print('PRESS 4 TO UPDATE MEDICINE COST')

print('PRESS 5 TO SELL MEDICINE')

print('PRESS 6 TO CHECK AVAILABILITY')

print('PRESS 7 TO DISPOSE EXPIRED MEDICINE')

print('PRESS 8 TO SEARCH EXPIRED MEDICINE BY NAME')

print('PRESS 9 TO VIEW QUANTITY,SELL,BALANCE GRAPHICALLY')

print('PRESS 10 TO CLOSE THE APPLICATION')

choice=int(input('ENTER YOUR CHOICE : '))

if(choice==1):

Store()

elif(choice==2):

Search\_by\_Name()

elif(choice==3):

Search\_by\_Manu()

elif(choice==4):

Cost\_Update()

elif(choice==5):

Sell()

elif(choice==6):

Available()

elif(choice==7):

Dispose()

elif(choice==8):

Search\_Dispose()

elif(choice==9):

Graph()

else:

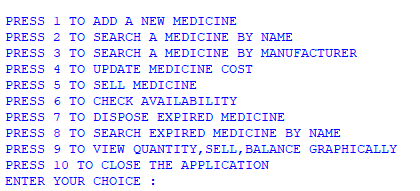
Close()

**MOTIVE**

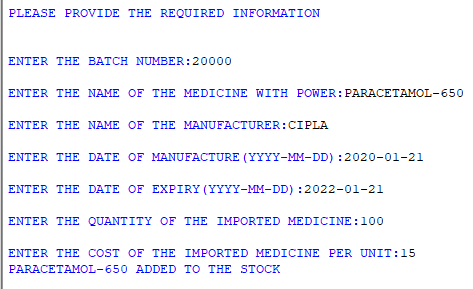
* To maintain the medicine stock details, sell medicine, update stock details, providing medicine amount enquiry by simple search technique.
* To dispose medicines which are expired and provide the facility to search the disposed medicines.
* To display the amount, sold amount, balance amount of a particular medicine by graphical analysis technique.
* Globalized usage.

**SCREEN SHOTS OF EXECUTION**

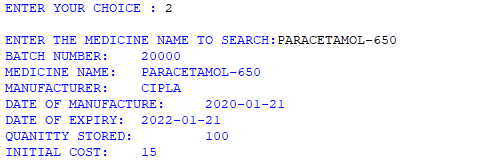
**MAIN MENU**



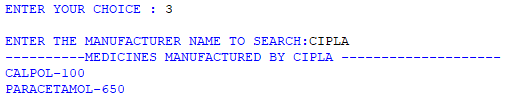
**ADDING A NEW MEDICINE**



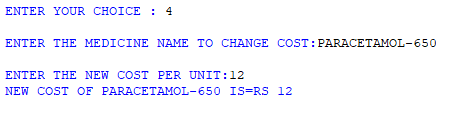
**SEARCHING MEDICINE BY NAME**



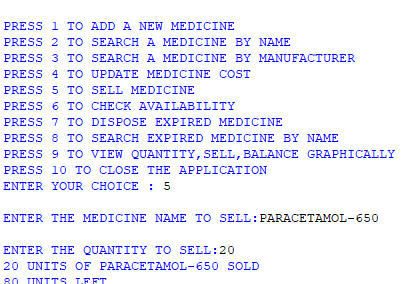
**SEARCHING MEDICINE BY MANUFACTURER**



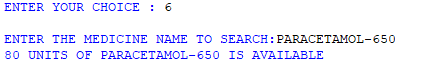
**UPDATING MEDICINE COST**



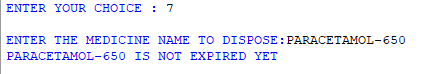
**SELLING MEDICINE**



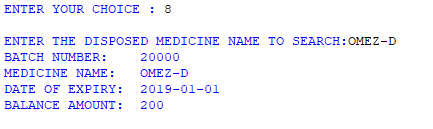
**CHECKING AVAILABILITY**



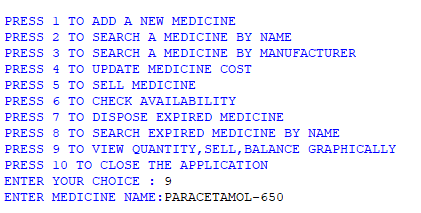
**DISPOSING MEDICINES**

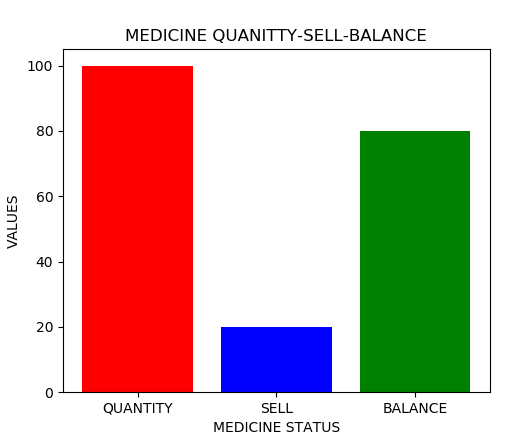


**SEARCHING EXPIRED MEDICINE BY NAME**



**VIEW QUANTITY,SELL,BALANCE GRAPHICALLY**



****

**BIBLIOGRAPHY**

**BOOKS:**

* COMPUTER SCIENCE WITH PYTHON- BY SUMITA ARORA
* COMPUTER SCIENCE WITH PYTHON-BY PREETI ARORA
* PYTHON COOKBOOK

**WEBSITES:**

* www.geeksforgeeks.org
* <https://docs.python.org/3/>
* [https://www.w3schools.com/python/](https://www.w3schools.com/python/python_strings.asp)

**LIMITATIONS**

* The project has no provision to calculate annual turnover of the medicine unit.
* The project does not incorporate the provision of GST Calculation.
* The project does not have the facility to take care of the medicines which are to be refunded i.e. there is no mechanism to keep the account of the refunded medicines.