**INDEX**

|  |  |
| --- | --- |
| **S. No** | **Topic** |
| **1.** | **INTRODUCTION** |
| **2.** | **LEARNING OUTCOME** |
| **3.** | **SYSTEM DESIGN** |
| **4.** | **DATABASE DESIGN** |
| **5.** | **INTERFACE DESIGN** |
| **6.** | **SOURCE CODE** |
| **7.** | **INPUT/OUTPUT SCREENSHOT** |
| **8.** | **REPORTS GENERATED** |
| **9.** | **FURTURE ENCHANCEMENT** |

1. **INTRODUCTION**

**Online Pharmaceutical Delivery Application**



CURAXIN provides you with medical information which is curated, written and verified by experts, accurate and trustworthy. Our experts create high-quality content about medicines, diseases, lab investigations, Over-The-Counter (OTC) health products, Ayurvedic herbs/ingredients, and alternative remedies. You can also order Ayurvedic, Homeopathic and other Over-The-Counter (OTC) health products. Your safety is our top priority. All products displayed on CURAXIN are procured from verified and licensed pharmacies.

At CURAXIN, we make a wide range of prescription medicines and other health products conveniently available all across India. Even second and third tier cities and rural villages can now have access to the latest medicines. Since we also offer generic alternatives to most medicines, online buyers can expect significant savings. With an aim to eradicate fake and ineffective medicines, and supply high-quality medicines in India, According to WHO research, every 1 or 2 in 10 medicines are proven to be adulterated in low/medium income countries like India and CURAXIN aspires to bring about a change in this statistic. To encourage and elevate transparency in the functioning of the pharmaceutical industry, CURAXIN has been successfully contributing in providing genuine and unadulterated medicines since its inception. Welcome to a seamless and impeccable shopping experience!

1. **LEARNING OUTCOME**

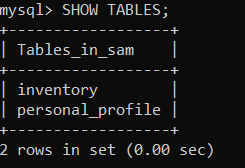
* Connecting to MySQL from Python.
* Applying MySQL queries for database definition and manipulation.
* With the help of Python, managing the front end and presenting the data stored in MySQL tables in an organized and meaningful manner.
* Application of conditional and looping statements, functions, lists, tuples, dictionaries, MySQL functions and other concepts into building an inventory management system!
* Moreover, writing such an extensive code helped in improving my logical reasoning, problem-solving ability and creativity while improving my coding skills!

1. **SYSTEM DESIGN**

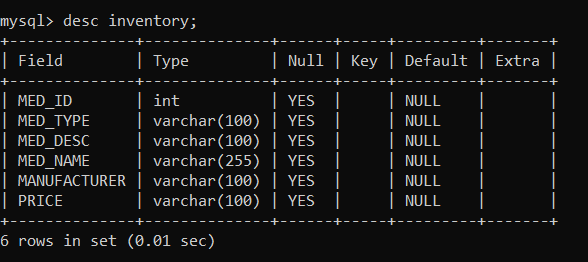
**Database name:**



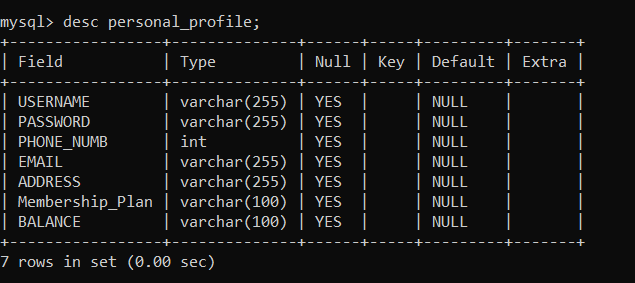
**Tables in Final Project:**

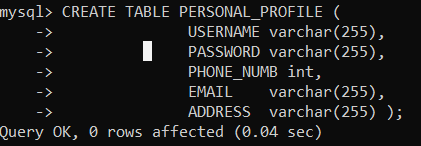


**Inventory Table:**



**Personal Profile table:**





**ONLINE PHARMACEUTICAL APPLICATION**

**CURAXIN**

**ENTITIES**

**-Administrator**

**-User**

**FUNCTIONAL AREAS**

### User

* **Profile**: To get the authentication, customers need to fill some of the required personal details such- an email address, or/and phone number.
* **Select the medicine**: users can choose the necessary medicine and select them or save them for the future.
* **Medicine details**: users can read a detailed prescription of each medication, see the manufacture, etc.
* Membership Plan: Enjoy exclusive member benefits and update and check your Plan. Get discount updates.
* Balance: Check balance on your Membership Wallet.

### Admin

* **Inventory management**: an admin keeps track of the inventory, orders, and medicine storing properly, gets notified about expiration dates and/or low stock by preventing medicine shortages and avoids confusion.
* **Market Price**: an admin can see the recent change in price of medicine, information with every single detail.
* **Manage Users**: an administrator is responsible for managing features of users.

1. **DATABASE DESIGN**

**MEDICINE/INVENTORY TABLE (MASTER TABLE)**

* MEDICINE ID
* MEDICINE NAME
* MEDICINE DESCRIPTION
* MEDICINE TYPE
* MANUFACTURER
* PRICE

**USER PERSONAL DETAILS TABLE (MASTER TABLE)**

* USERNAME
* PASSWORD
* PHONE NO.
* EMAIL ADDRESS
* ADDRESS
* MEMBERSHIP PLAN
* BALANCE

**MENU DESIGN**

**USER**

1. USER INTERFACE
   * 1. NAME
     2. MEDICINE ID
     3. MEDICINE PRICE
     4. SEARCH BY MEDICINE NAME
     5. SEARCH BY DESC.
     6. SEARCH BY MEDICINE ID
     7. SEARCH BY MEDICINE
     8. MANUFACTURER
     9. VIEW PROFILE
     10. EDIT PROFILE

**ADMIN**

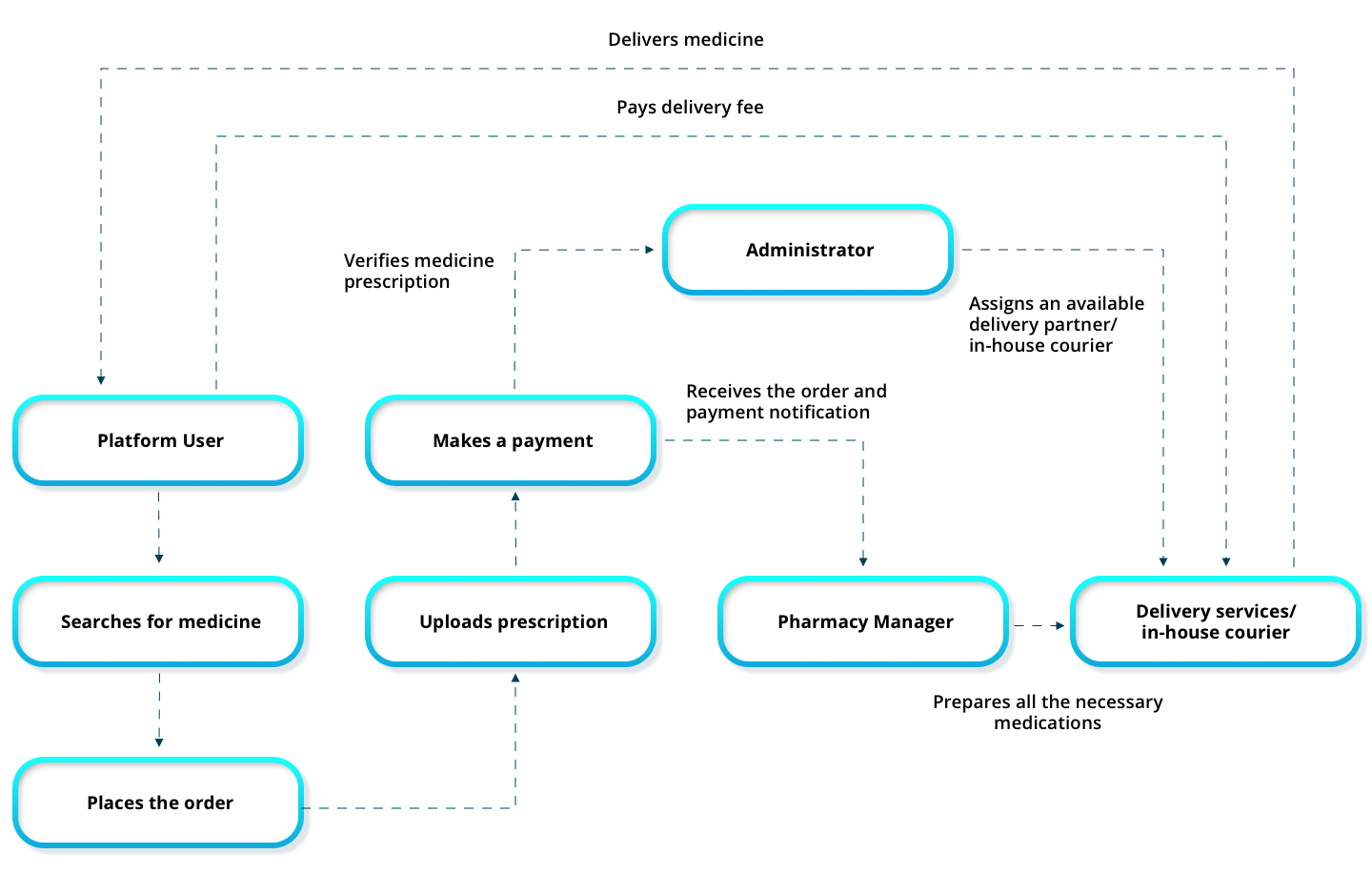
**INVENTORY**

* ADD DRUGS
* REMOVE DRUGS

**USER MANAGEMENT**

* VIEW ALL ACTIVE USERS
* REMOVE USER PROFILE

1. **INTERFACE DESIGN**



1. **SOURCE CODE**

print(''' ✦•················•»——————⋆◦★◦⋆——————«•····················•✦

»»----------<:~ 『CURAXIN』 ~:>----------««

❈─────•✦•❅•✦•─────❈ ''')

import random

import re

import mysql.connector

import datetime

con=mysql.connector.connect(host="localhost",

user="root",

password="12345",

database='sam')

cur=con.cursor()

flag=0

MNAME="SamNotFound"

while flag<=0:

A=int(input("\nPRESS 1 TO LOG IN TO YOUR ACCOUNT:\

\nPRESS 2 TO JOIN US:\

\nPRESS 3 TO LOG IN AS ADMIN:\n"))

**if A==1:**

D=input("ENTER YOUR E-MAIL ID:")

query="select USERNAME from PERSONAL\_PROFILE where EMAIL = '%s'"%(D)

cur.execute(query)

data=cur.fetchall()

if data == [] :

print("Incorrect Email\_ID\

\nJOIN NOW SIGN UP WITH US TODAY :)")

flag-=1

else:

print("Correct Email\_ID")

UPAS=input("ENTER YOUR PASSWORD:")

PASS="select PASSWORD from PERSONAL\_PROFILE where EMAIL = '%s'"%(D)

cur.execute(PASS)

Data=cur.fetchall()

for i in Data:

def convert(i):

t= ''.join(i)

return t

UPASS=convert(i)

if UPASS!= str(UPAS):

print("Incorrect Password\

\nJOIN NOW SIGN UP WITH US TODAY:)")

flag-=1

else:

flag+=1

print("Correct Password")

for i in data:

def convert(i):

t= ''.join(i)

return t

s = convert(i)

S = s

print (S,"Welcome to CURAXIN")

**elif A==2:**

p=0

while p>=0:

N=input("ENTER YOUR NAME:")

if str(N) == str(MNAME):

p+=1

print("LOGIN USING ADMIN PORTAL Mr.KIM-JONG-UN")

break

else:

print("VALID USERNAME")

p-=1

PN=int(input("ENTER YOUR Phone no.:"))

if len(str(PN))!=8:

print("Invalid Phone number")

p+=0

else:

EA=input("ENTER YOUR Email Address:")

SEA= ( r'\b[a-z0-9.\_%+-]+@[a-z0-9]+\.[a-z]{2,}\b')

def check(EA):

if (re.fullmatch(SEA,EA)):

print("Valid Email ID")

PW=input("ENTER YOUR NEW PASSWORD(\nONLY 8 CHARACTER\

\nAt least 1 number or digit between [0-9]\

\nAt least 1 character from [ \_ or @ or $ or &]:-")

password = PW

if(len(password)<8)and\

not re.search("[0-9]", password) and\

not re.search("[\_@$&]", password) and\

not re.search("\s", password):

p+=0

print("Not a Valid Password")

else:

print("Valid Password")

CYP=input("JOIN US NOW WITH(ENTER Rs AND THE PRICE ONLY)\

\nJOIN CURAXIN FOR 1 MONTH AT Rs399\-\

\nJOIN CURAXIN FOR 2 MONTH AT Rs799\-\

\nJOIN CURAXIN FOR 3 MONTH AT Rs999\-\

\nJOIN CURAXIN FOR 6 MONTH AT Rs1550\-\

\nJOIN CURAXIN FOR ANNUAL MEMBERSHIP AT Rs2749\-:-")

AM=input("Enter Your Address:")

BL=input("Enter Balance Amount:")

query=("Insert into personal\_profile values('{}','{}',{},'{}','{}','{}','{}')".format(N,PW,PN,EA,AM,CYP,BL))

cur.execute(query)

con.commit()

print(N,"Welcome to CURAXIN")

else:

print("Invalid Email ID")

check(EA)

**elif A==3:**

ANAME=input("Enter your name:")

if str(ANAME) != str(MNAME):

print ("INCORRECT USERNAME")

flag-=1

break

**else:**

AEMAIL=input("ENTER YOUR EMAIL ADDRESS:-")

ADEMAIL="select EMAIL from PERSONAL\_PROFILE where USERNAME = '%s'"%(ANAME)

cur.execute(ADEMAIL)

DaTa=cur.fetchall()

print(DaTa)

for i in DaTa:

def convert(i):

t= ''.join(i)

return t

ADMAIL = convert(i)

print(ADMAIL)

if AEMAIL !=str(ADMAIL) :

print("Incorrect Email\_ID")

flag-=1

break

else:

print("Correct Email\_ID")

ADPASS=input("ENTER YOUR PASSWORD:-")

ADPAS="select PASSWORD from PERSONAL\_PROFILE where EMAIL = '%s'"%(AEMAIL)

cur.execute(ADPAS)

dAtA=cur.fetchall()

for i in dAtA:

def convert(i):

t= ''.join(i)

return t

SS = convert(i)

if SS != str(ADPASS):

print("Incorrect Password")

flag-=1

break

else:

flag+=1

print("Correct Password")

print (MNAME,"Welcome to CURAXIN")

k=0

while k<=0:

Y=int(input("CHOOSE THE FOLLOWING OPTIONS:\

\nADD DRUGS= 1 REMOVE DRUGS= 2\

\nVEIW ALL = 3 REMOVE USER PROFILE= 4\n"))

**if Y==1:**

MN=input("ENTER MEDICINE NAME:")

MID=input("ENTER THE MEDICINE ID:")

DM=input("DESC OF MEDICINE:")

TM=input("ENTER THE TYPE OF MEDICINE:")

MD=input("ENTER THE MANUFACTURER OF THE DRUGS:")

PR=input("Enter Medicine Price:")

AM="insert into inventory values({},'{}','{}','{}','{}','{}')".format(MID,TM,DM,MN,MD,PR)

cur.execute(AM)

con.commit()

print("MEDICINE ADDED SUCCESSFULLY")

k+=0

**elif Y==2**:

MNR=input("ENTER MEDICINE NAME WHICH YOU WANT TO REMOVE:")

DMN="delete from inventory where MED\_NAME ={}".format(MNR)

cur.execute(DMN)

print("MEDICINE REMOVED SUCCESSFULLY")

**elif Y==3:**

VU="select \* from personal\_profile"

cur.execute(VU)

OP=cur.fetchmany()

print(OP)

**elif Y==4:**

UN=input("ENTER USERNAME:-")

RU="delete from personal\_profile where USERNAME={}".format(UN)

cur.execute(RU)

else:

print("CHOOSE AN OPTION")

flag-=1

f=0

while f<=0:

x=int(input("\nEnter number to choose the options\

\nMedicine Manufacturer=1\

\nMedicine Type=2\

\nView Profile=3\

\nSearch Medicine=4\

\nSearch Med ID=5\n"))

**if x==1:**

C=input("Alkem, Apollo, Astellas, Abbott\

\nAmgen, AstraZeneca, Aventis, Biocon\

\nBiogen,Bristol-Myers Squibb, Cadila,,Cipla\

\nDivi's Lab, Dr.Reddy's, Eli Lilly, GlaxoSmithKline\

\nJohnson and Johnson, Novo Nordisk, Novartis, Pfizer\

\nPiramal, Roche, Schering-Plough, Sinopharm\

\nSun Pharma,Takeda, Teva, Torrent\n")

SC="select \* from Inventory where MANUFACTURER = '{}'".format(C)

cur.execute(SC)

SBC=cur.fetchall()

for BC in SBC:

print(BC)

**elif x==2:**

R=input(" | Ayurveda | Capsules | Drops | Drugs |\

\n | Homoeopathy | Inhalers | Injections | Liquid |\

\n | Patches | Suppositories | Tablet |\n")

SC="select \* from Inventory where MED\_TYPE = '{}'".format(R)

cur.execute(SC)

SBC=cur.fetchall()

for BC in SBC:

print(BC)

**elif x==3:**

q=0

while q>=0:

PRO=int(input("1--EDIT USER DATA\

\n2--VIEW USER PROFILE\n"))

if PRO==1:

UpdatePROF=int(input("What would you like to change: 1--NAME\

\n2--PHONENUMBER\

\n3--EMAIL\

\n4--PASSWORD\

\n5--Membership\n"))

**if UpdatePROF==1:**

CNAME=input("ENTER YOUR CURRENT NAME:")

NNAME=input("ENTER YOUR NEW NAME:")

PROF="update PERSONAL\_PROFILE set USERNAME='{}'where USERNAME='{}'".format(NNAME,CNAME)

cur.execute(PROF)

con.commit()

print("NAME UPDATED")

break

**elif UpdatePROF==2:**

CPN=input("ENTER YOUR CURRENT PHONENUMBER:")

NPN=input("ENTER YOUR NEW PHONENUMBER:")

if len(str(NPN))!=8:

print("Invalid Phone number")

q+=0

else:

q-=0

print("Valid Phone number")

PHONE="update PERSONAL\_PROFILE set PHONE\_NUMB={}where PHONE\_NUMB={}".format(NPN,CPN)

cur.execute(PHONE)

con.commit()

print("PHONENUMBER UPDATED")

break

**elif UpdatePROF==3:**

CEA=input("ENTER YOUR CURRENT EMAIL ADDRESS:")

NEA=input("ENTER YOUR NEW EMAIL ADDRESS:")

def check(NEA):

SEA= (r'\b[a-z0-9.\_%+-]+@[a-z0-9]+\.[a-z]{2,}\b')

if (re.fullmatch(SEA,NEA)):

print("Valid Email ID")

q-=0

EADD="update PERSONAL\_PROFILE set EMAIL={}where EMAIL={}".format(NEA,CEA)

cur.execute(EADD)

con.commit()

print("EMAIL ADDRESS UPDATED")

else:

print("Invalid Email ID")

q+=0

check(NEA)

break

**elif UpdatePROF==4:**

CPASS=input("ENTER YOUR CURRENT PASSWORD:")

EPASS=input("ENTER YOUR NEW PASSWORD(\nONLY 8 CHARACTER\

\nAt least 1 number or digit between [0-9]\

\nAt least 1 character from [ \_ or @ or $ or &]:-")

if(len(EPASS)<8)and\

not re.search("[0-9]", EPASS) and\

not re.search("[\_@$&]", EPASS) and\

not re.search("\s", EPASS):

q+=0

print("InValid Password")

else:

q-=0

print("Valid Password")

PAS="update PERSONAL\_PROFILE set PASSWORD='{}'where PASSWORD='{}'".format(NPASS,CPASS)

cur.execute(PAS)

con.commit()

print("PASSWORD UPDATED")

break

**elif UpdatePROF==5:**

q-=0

CP=input("ENTER YOUR PLAN NAME:")

NP=input("CHOOSE YOUR PLAN(ENTER Rs AND THE PRICE ONLY)\

\nJOIN CURAXIN FOR 1 MONTH AT Rs399\-\

\nJOIN CURAXIN FOR 2 MONTH AT Rs799\-\

\nJOIN CURAXIN FOR 3 MONTH AT Rs999\-\

\nJOIN CURAXIN FOR 6 MONTH AT Rs1550\-\

\nJOIN CURAXIN FOR ANNUAL MEMBERSHIP AT Rs2749\-\n")

PLAN="update PERSONAL\_PROFILE Membership\_Plan='{}'where Membership\_Plan='{}'".format(NP,CP)

cur.execute(PLAN)

con.commit()

print("PLAN UPDATED")

else:

print("CHOOSE AN OPTION")

**elif PRO==2:**

q-=0

**if A==1:**

q-=0

PROF="select \* from PERSONAL\_PROFILE where USERNAME = '%s'"%(S)

cur.execute(PROF)

PROFILE=cur.fetchall()

for i in PROFILE:

print(i)

**elif A==2:**

q-=0

PROF="select \* from PERSONAL\_PROFILE where USERNAME = '%s'"%(N)

cur.execute(PROF)

PROFILE=cur.fetchall()

for i in PROFILE:

print(i)

**elif PRO==3:**

f+=0

else:

q+=0

print("CHOOSE AN OPTION")

**elif x==4:**

search=input("Search a MEDICINE NAME:")

FE=(search+"%")

F=("select \* from inventory where MED\_NAME LIKE '{}'".format(FE))

cur.execute(F)

FU=cur.fetchall()

for FUu in FU :

print(FUu)

**elif x==5:**

search=input("Search MEDICINE\_id")

FE=(search+"%")

F=("select \* from inventory where MED\_ID LIKE '{}'".format(FE))

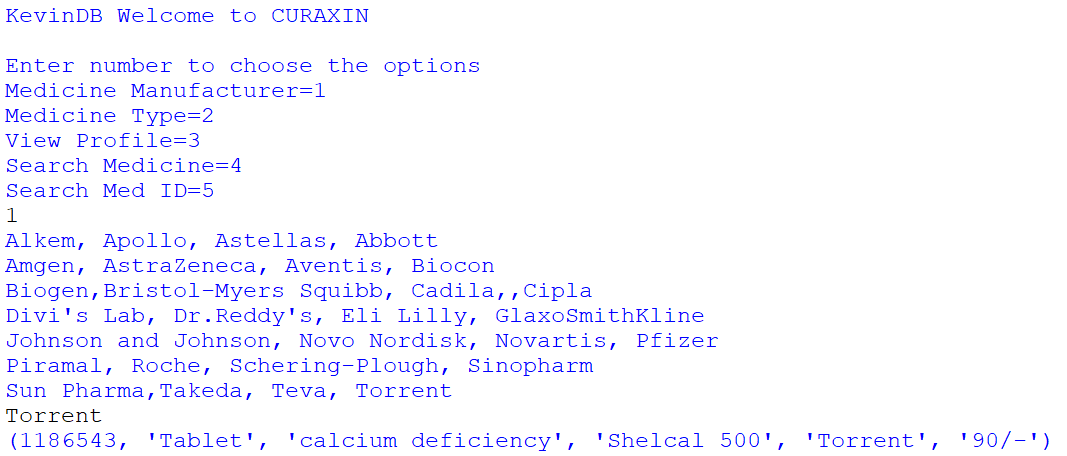
cur.execute(F)

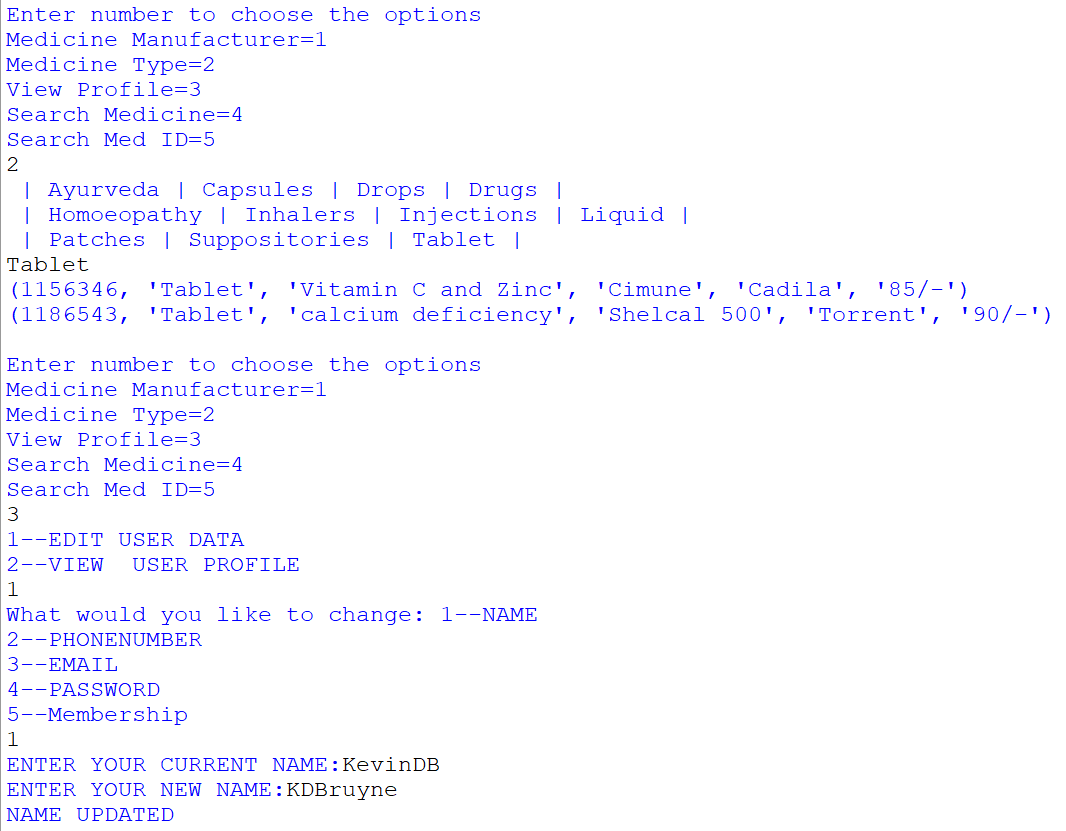
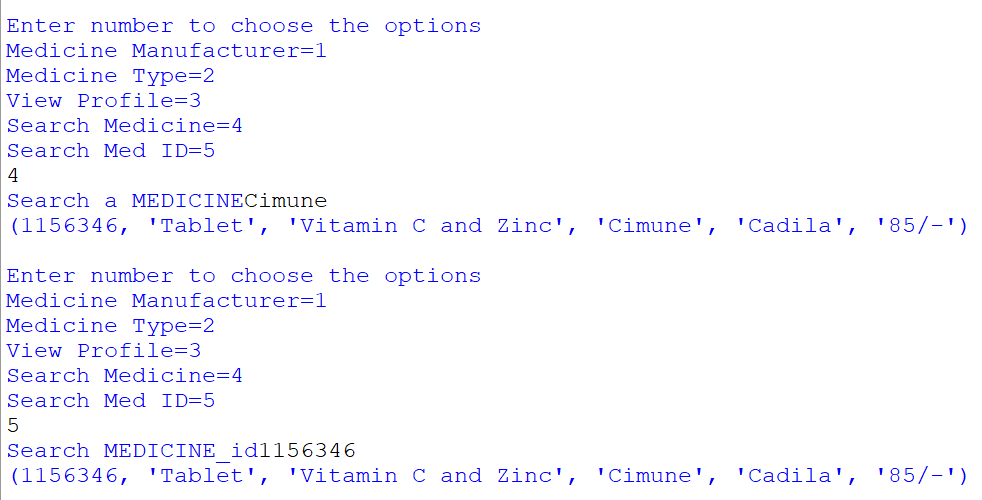
FU=cur.fetchall()

for FUu in FU :

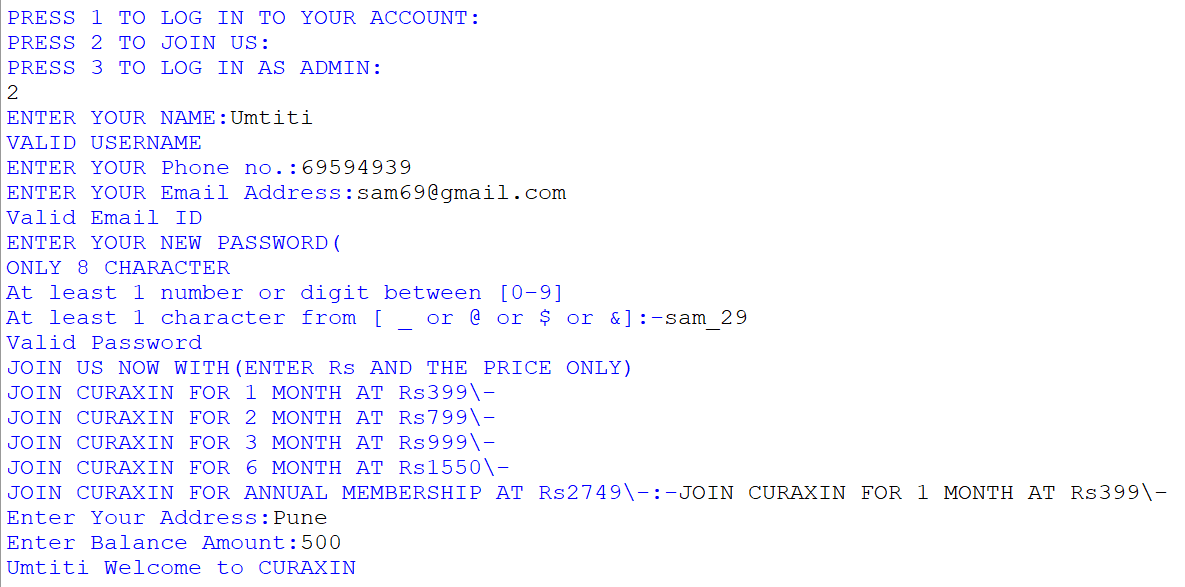
print(FUu)

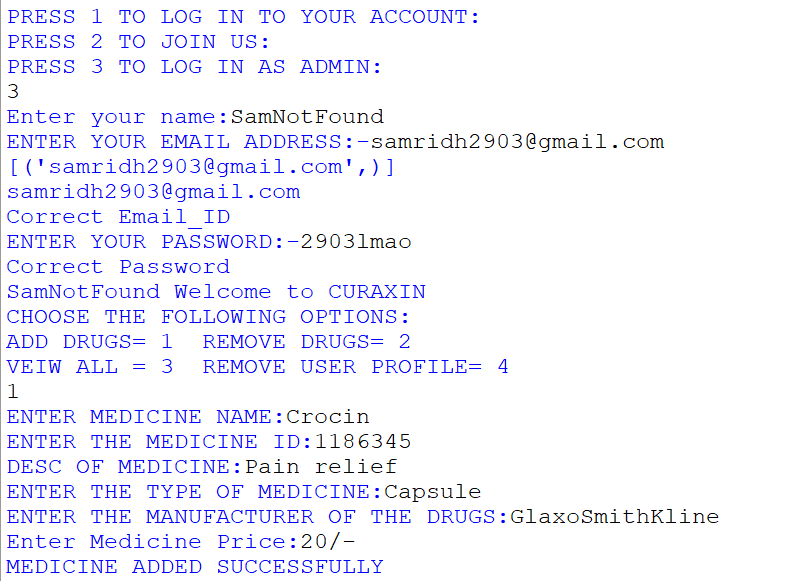
1. **INPUT/OUTPUT**

****

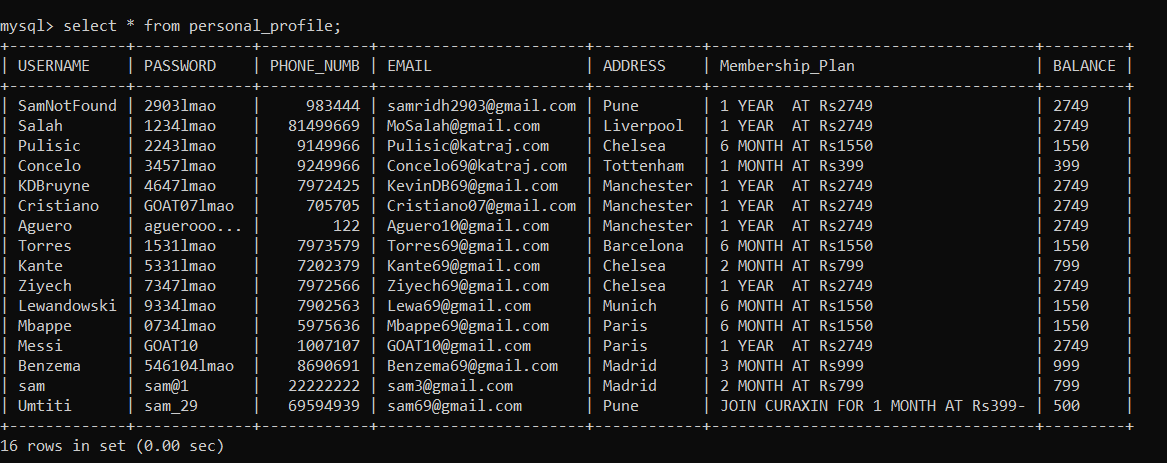
****

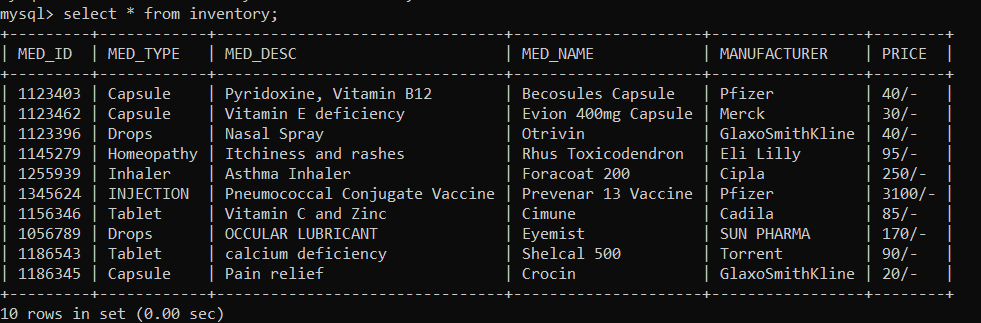
****

****

****

**9. REPORTS GENERATED**

**Personal Profile**

**Inventory**

**10.Conclusion and Future Scope**

* Detailed information gathering has to be done. Without that the purpose for using the software won’t be satisfied properly.
* However, it can give good profits in the long run.
* Implementing the software requires change in the business practices.
* Efficient organization of all knowledge is the analysis company and easy analysis access and retrieval of information is possible.
* In this project we can also include BAR CODE facility using the bar code reader, which will detect the expiry date and the other information about the related medicines.
* Company using this software will always be able to plan in future and always be aware of their financial position in the market.
* It leads to streaming of business processes.
* Help in digitalization of country’s economy with cashless transaction.
* Minimum contact and fast delivery within 24 hours during this Pandemic, ensuring we can save lives.