तत् त्वं पूषन् अपावृणु

केन्द्रीय विद्यालय संगठन

COMPUTER SCIENCE PROJECT

Kendriya Vidyalaya Badarpur

Index

ACKNOWLEDGMENT

This project consumed a huge amount of work, research, and dedication. Still, implementation would not have been possible if we did not have the support and expertise of many individuals. Therefore, we would like to extend our sincere gratitude to all of them.

We would like to express our special thanks of gratitude to our teacher **Mr. Ateet Bhatt Sir** (PGT Computer Science) who gave us the golden opportunity to do this wonderful project.

Secondly, we would also like to thank our parents who supported us a lot in finalizing this project within the limited time frame.

CERTIFICATE

This is to certify that

- Shubham Dev
- Ayush Vishwakarma
- Vasudev Sharma
- Kaushal Meena
- Mohit Meena

Students of Class XII-A have successfully completed the project on

“ Hospital Database Management System “

in the subject Computer Science (Code - 083) for the purpose of Practical Examination of Class XII to be held in Kendriya Vidyalaya Badarpur New Delhi

Examiner –

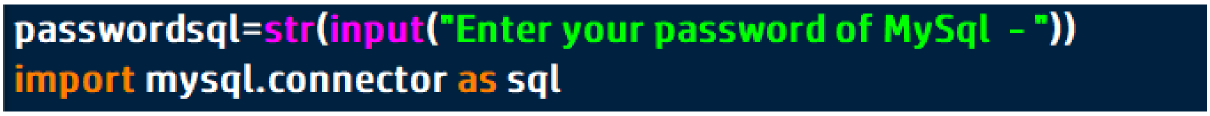
Signature –

SYNOPSIS

With the Python Code that we have written, what we want to achieve is that if a Hospital uses any relational database management system like MySQL for the purpose of storing and accessing data then they can use our code so that they do not have to write very long queries to do even simple things like seeing data of a patients or doctors. We have made it very user friendly and the person does not need to have any prior knowledge of neither SQL nor of Python. The person can see the data of all patients/doctors or they can see the data of a particular patient/doctor. They can also add records into the table. The content of the table can also be updated without the need to enter any complicated SQL query. One the most unique thing in program is that the person can have the data of the table in the form a .csv file or .txt file. We have integrated a lot of things that we have learnt into this code, we have also integrated some things which may not be part of our syllabus as this was research project.

ALGORITHM

1.



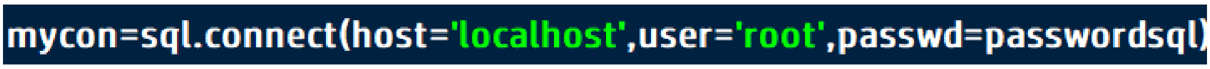
```
passwordsql=str(input("Enter your password of MySql  - "))
import mysql.connector as sql
```

Asks the user password of MySQL of his system.

Stores it in string format in passwordsql(variable)

Imports MySQL Connector

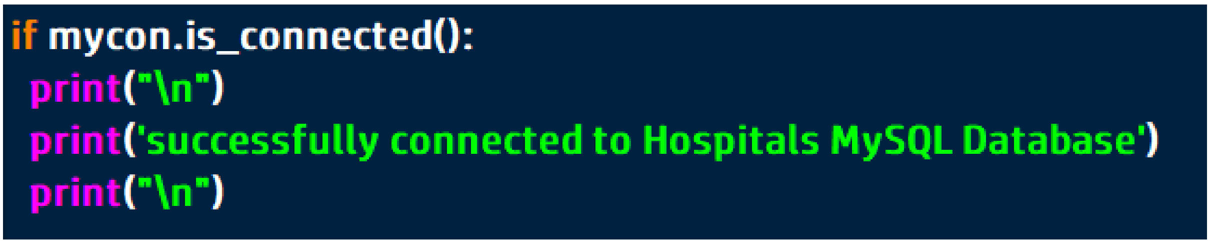
2.



```
mycon=sql.connect(host='localhost',user='root',passwd=passwordsql)
```

Connects to MySQL using a password stored in variable *passwordsql*

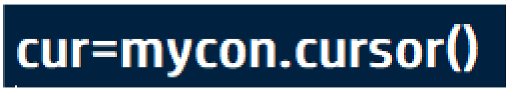
3.



```
if mycon.is_connected():
  print("\n")
  print('successfully connected to Hospitals MySQL Database')
  print("\n")
```

Tells the user if the connection is successful.

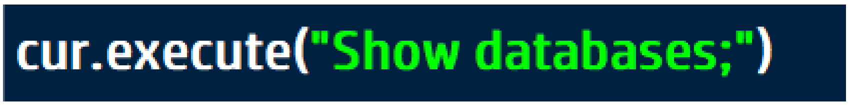
4.



```
cur=mycon.cursor()
```

A **Database Cursor** is created to retrieve the resultset. A DB Cursor allows us to traverse the result set row by row.

5.

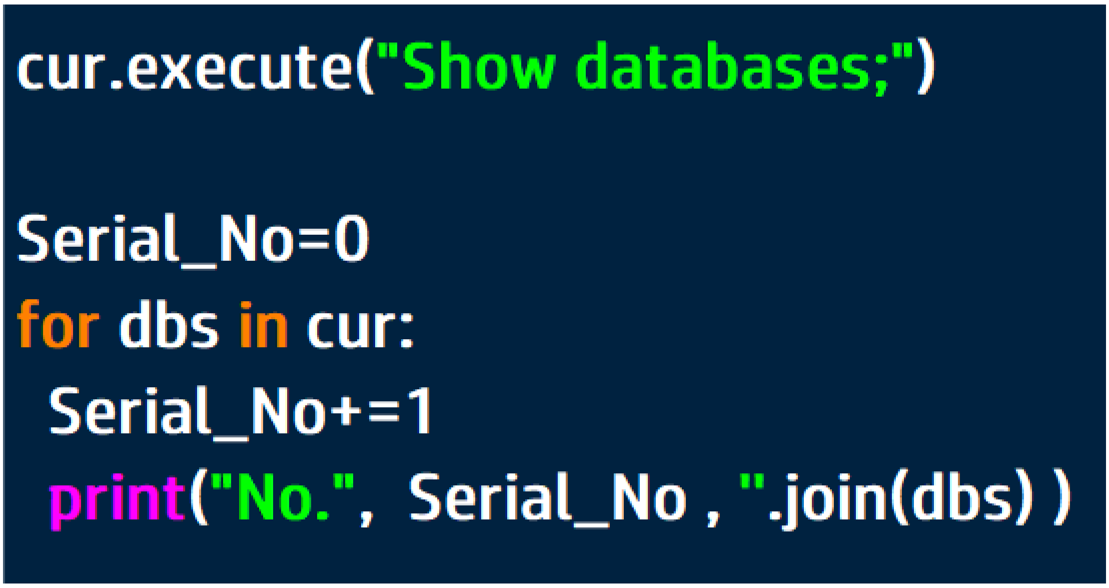


```
cur.execute("Show databases;")
```

Using execute() function with cursor object as per syntax

<cursorobject>.execute(<sql query string>)

6.

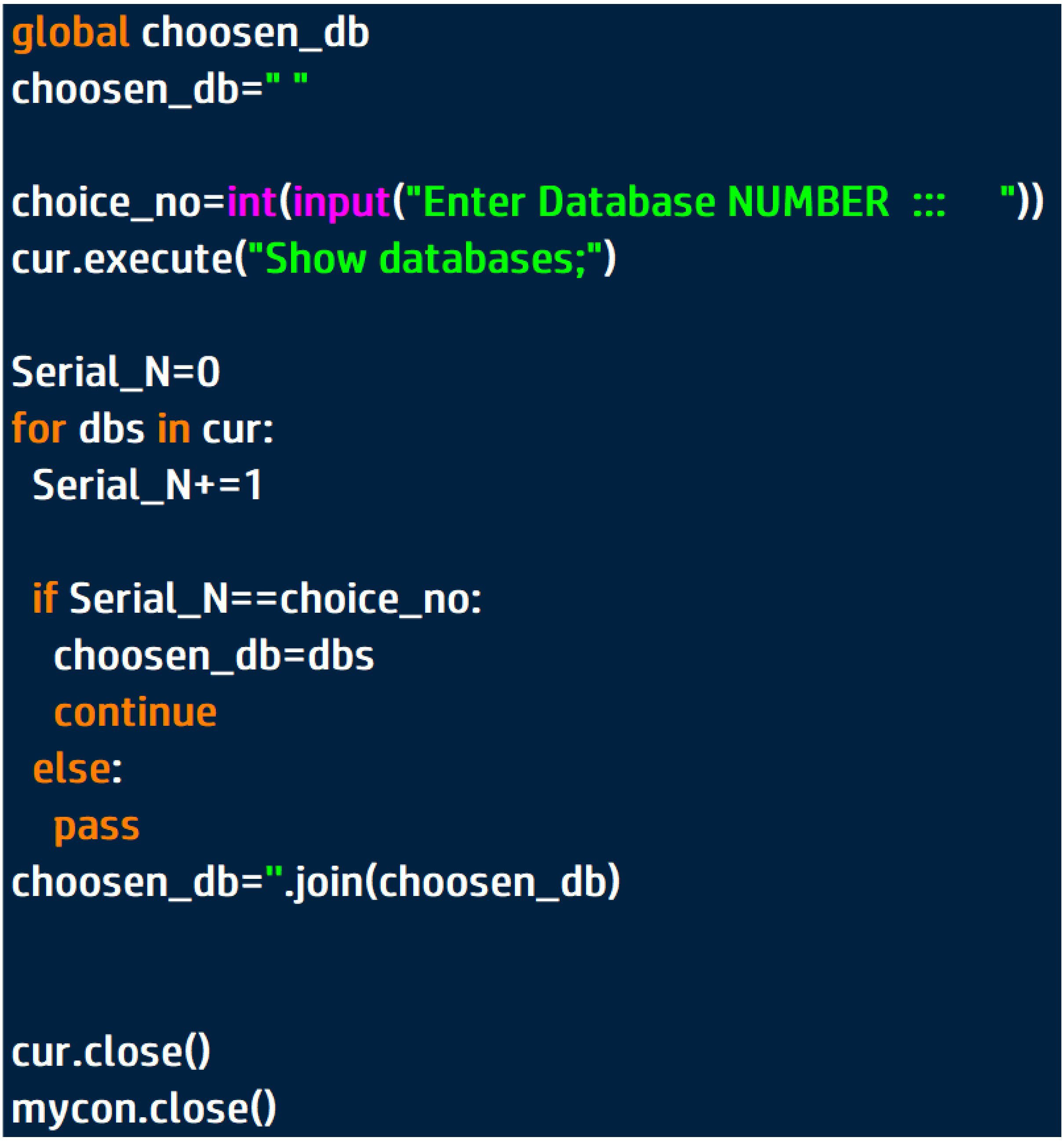


```
cur.execute("Show databases;")

Serial_No=0
for dbs in cur:
  Serial_No+=1
  print("No.",  Serial_No , ''.join(dbs) )
```

A **for loop** to get the **resultset** from cur and print it.

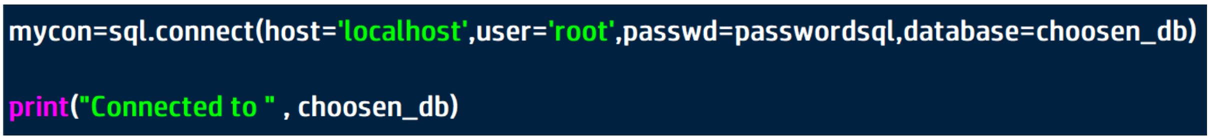
7.

A program through which the user has to input the database number. A for loop runs and that database is chosen whose serial number is entered by the user and database name is stored in the variable *choosen_db*. The MySQL connection is closed.



```
global choosen_db
choosen_db=" "

choice_no=int(input("Enter Database NUMBER  :::     "))
cur.execute("Show databases;")

Serial_N=0
for dbs in cur:
  Serial_N+=1

  if Serial_N==choice_no:
    choosen_db=dbs
    continue
  else:
    pass
choosen_db=''.join(choosen_db)



cur.close()
mycon.close()
```

8.



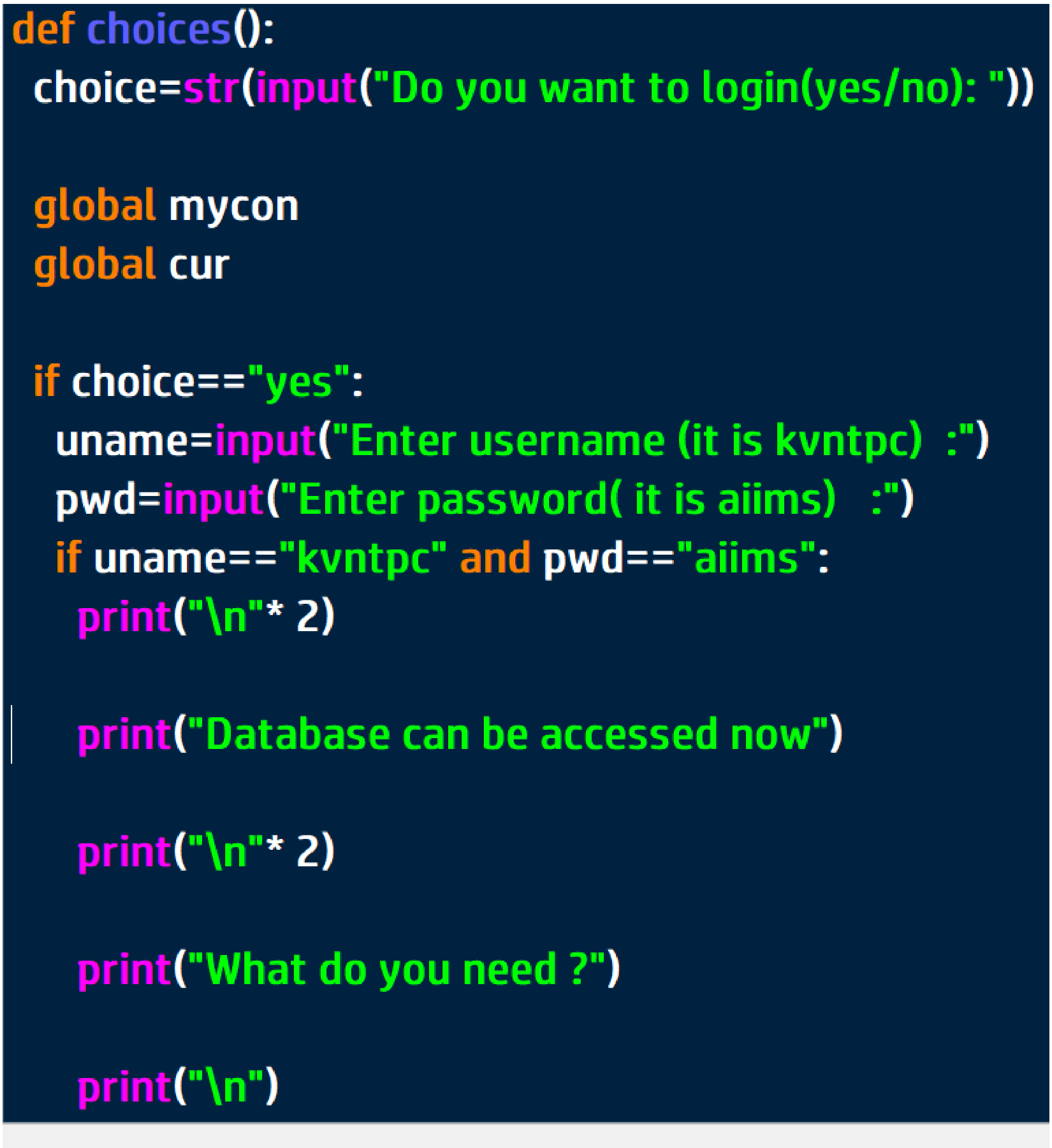
```
mycon=sql.connect(host='localhost',user='root',passwd=passwordsql,database=choosen_db)

print("Connected to " , choosen_db)
```

After disconnection from MySQL, Python again connects with MySQL but this time it connects with the database that the user had chosen.

9.

Our whole rest of the code is defined under a single function **choices().**

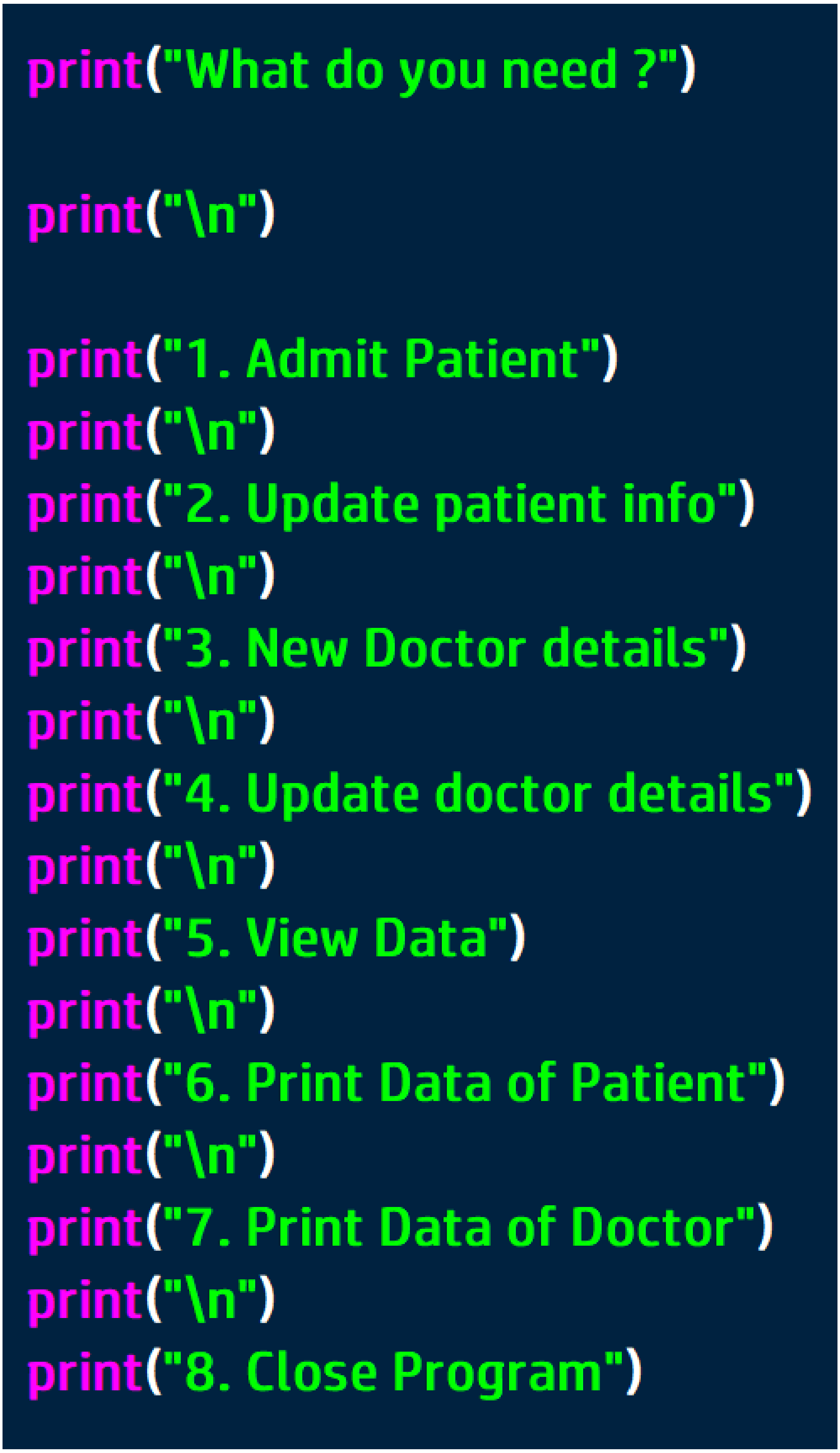
- This is a very unique thing about this program as it allows us to call the function again without the need to re-run the code and there is no need to run the program from start.



```
def choices():
  choice=str(input("Do you want to login(yes/no): "))

  global mycon
  global cur

  if choice=="yes":
    uname=input("Enter username (it is kvntpc)  :")
    pwd=input("Enter password( it is aiims)   :")
    if uname=="kvntpc" and pwd=="aiims":
      print("\n"* 2)

      print("Database can be accessed now")

      print("\n"* 2)

      print("What do you need ?")

      print("\n")
```

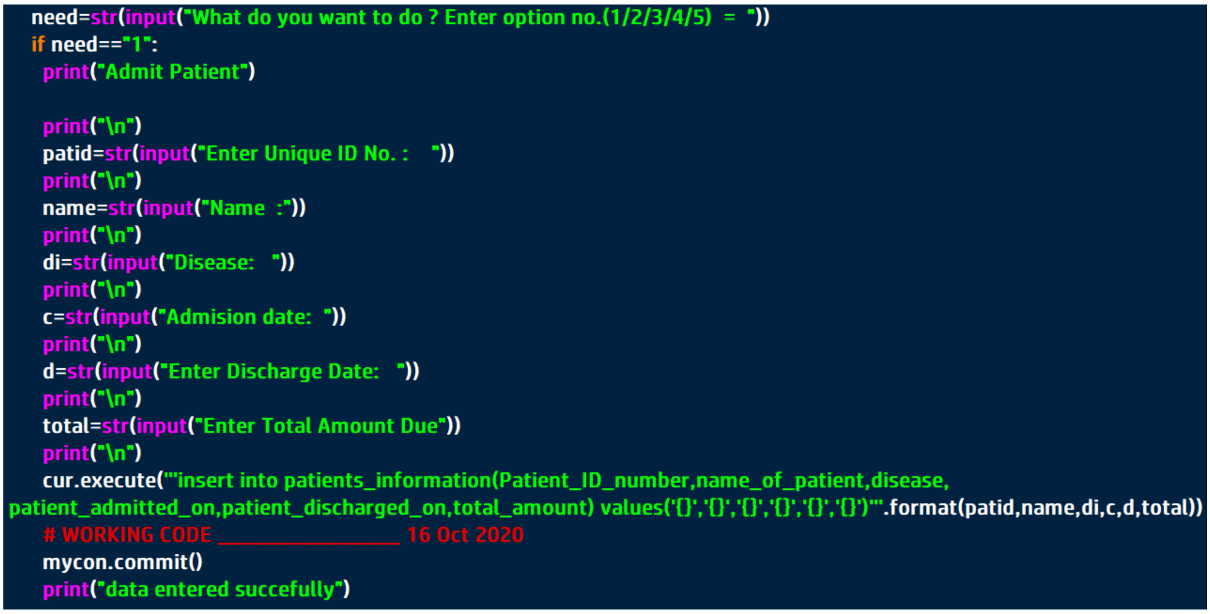
10.

The user is shown the different options that are available to him.



```
print("What do you need ?")

print("\n")

print("1. Admit Patient")
print("\n")
print("2. Update patient info")
print("\n")
print("3. New Doctor details")
print("\n")
print("4. Update doctor details")
print("\n")
print("5. View Data")
print("\n")
print("6. Print Data of Patient")
print("\n")
print("7. Print Data of Doctor")
print("\n")
print("8. Close Program")
```

11.



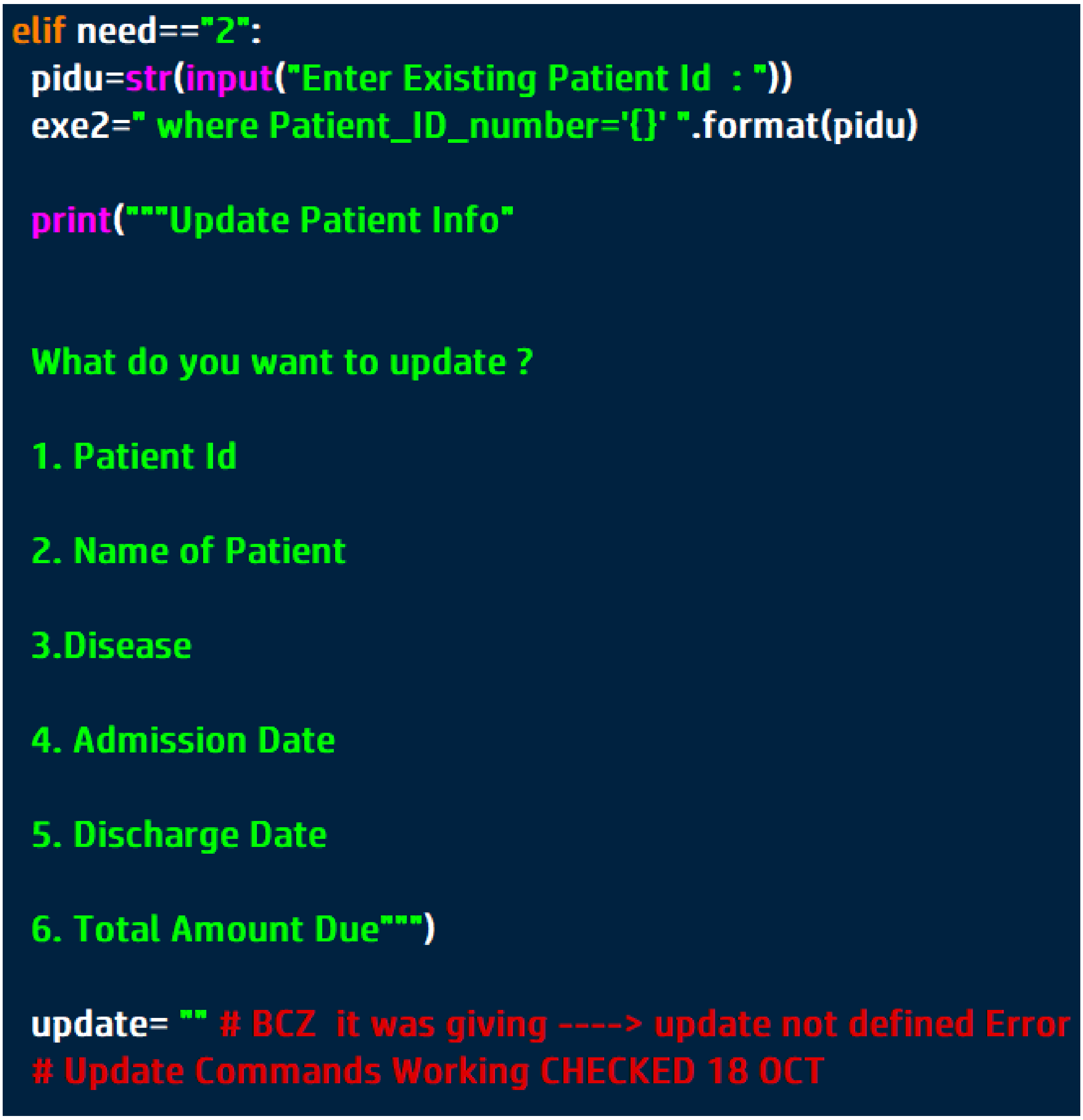
```
  need=str(input("What do you want to do ? Enter option no.(1/2/3/4/5)  =  "))
  if need=="1":
    print("Admit Patient")

    print("\n")
    patid=str(input("Enter Unique ID No. :    "))
    print("\n")
    name=str(input("Name  :"))
    print("\n")
    di=str(input("Disease:   "))
    print("\n")
    c=str(input("Admision date:  "))
    print("\n")
    d=str(input("Enter Discharge Date:   "))
    print("\n")
    total=str(input("Enter Total Amount Due"))
    print("\n")
    cur.execute('''insert into patients_information(Patient_ID_number,name_of_patient,disease,
patient_admitted_on,patient_discharged_on,total_amount) values('{}','{}','{}','{}','{}','{}')'''.format(patid,name,di,c,d,total))
    # WORKING CODE _________________ 16 Oct 2020
    mycon.commit()
    print("data entered succefully")
```

If the user enters 1 then the user is asked details about the patient which is stored in different variable. But to execute the SQL Command in which user has entered the details was problematic for us. So, we used **format() method** which is not in our syllabus of Class 12th. We researched about this on the internet. The **format() method** formats the specified values and insert them inside the string's placeholder.

The placeholder is defined using curly brackets: **{}** as can be seen in our code. The data entered is committed using **commit()**

12.

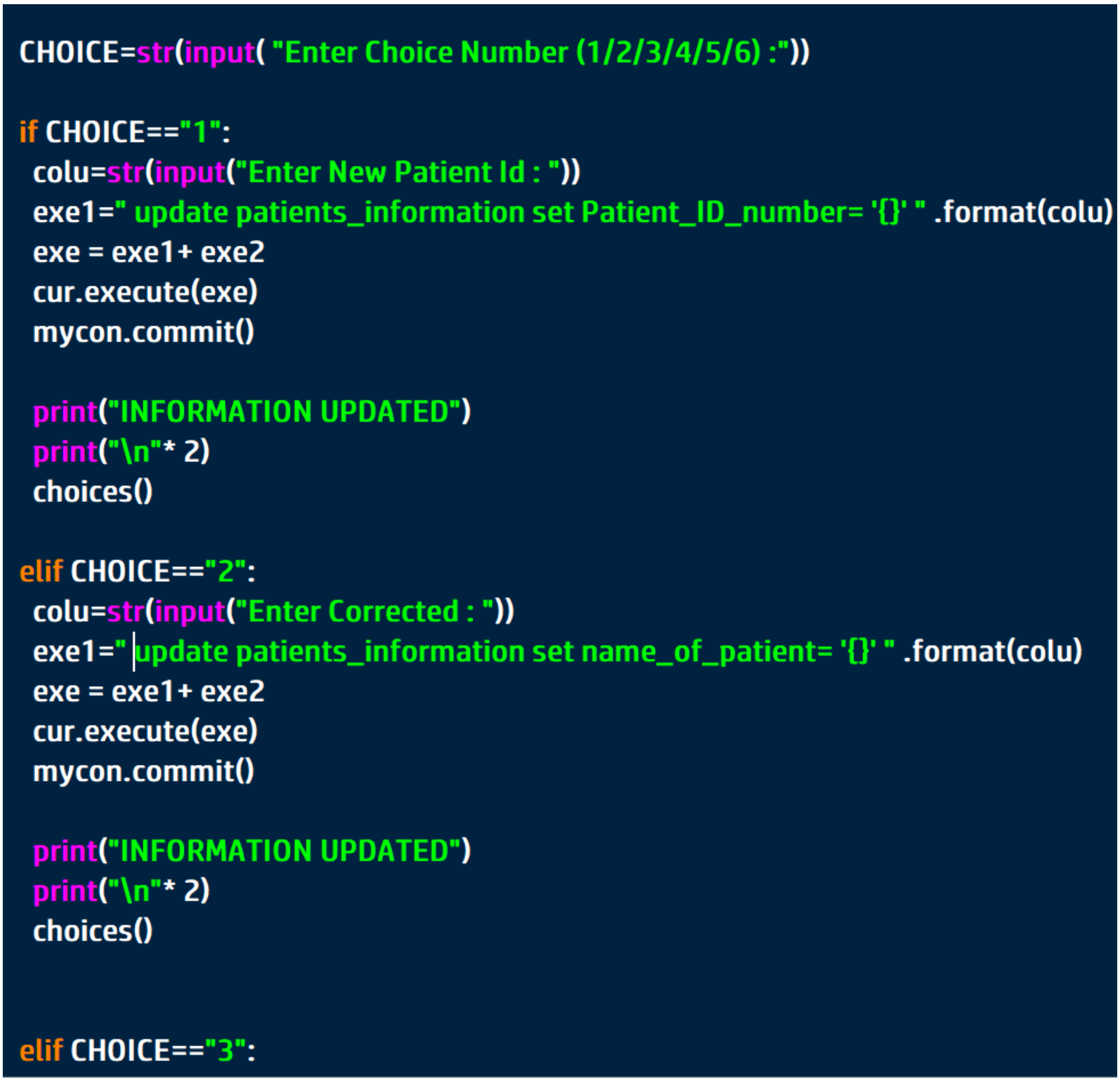
If the user inputs 2 to the variable *need* then this code will run. The user is asked to enter *Patient Id* and is stored in *exe2.* The user is asked to enter what part of the table Patient_Data that the user wants to change and that is stored in the variable **CHOICE**



```
elif need=="2":
  pidu=str(input("Enter Existing Patient Id  : "))
  exe2=" where Patient_ID_number='{}' ".format(pidu)

  print("""Update Patient Info"


  What do you want to update ?

  1. Patient Id

  2. Name of Patient

  3.Disease

  4. Admission Date

  5. Discharge Date

  6. Total Amount Due""")

  update= "" # BCZ  it was giving ----> update not defined Error
  # Update Commands Working CHECKED 18 OCT
```

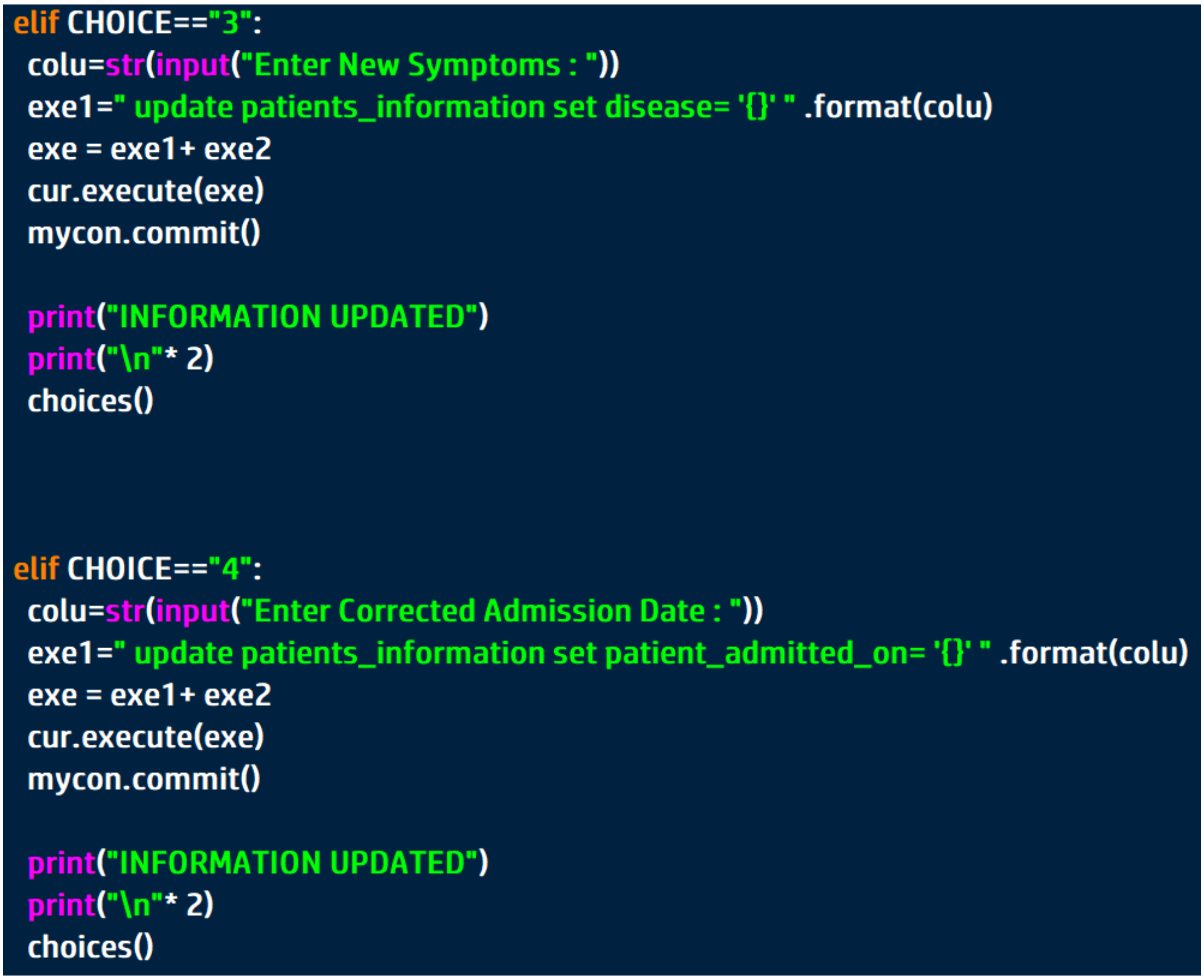
13.

An **if-elif** program is made here for the purpose. As the user has already entered the existing **Patient_ID_number** and the query of which is stored in the variable exe2. The user now enters the specific data that needs to be updated and that is integrated to a query stored in variable exe1 .

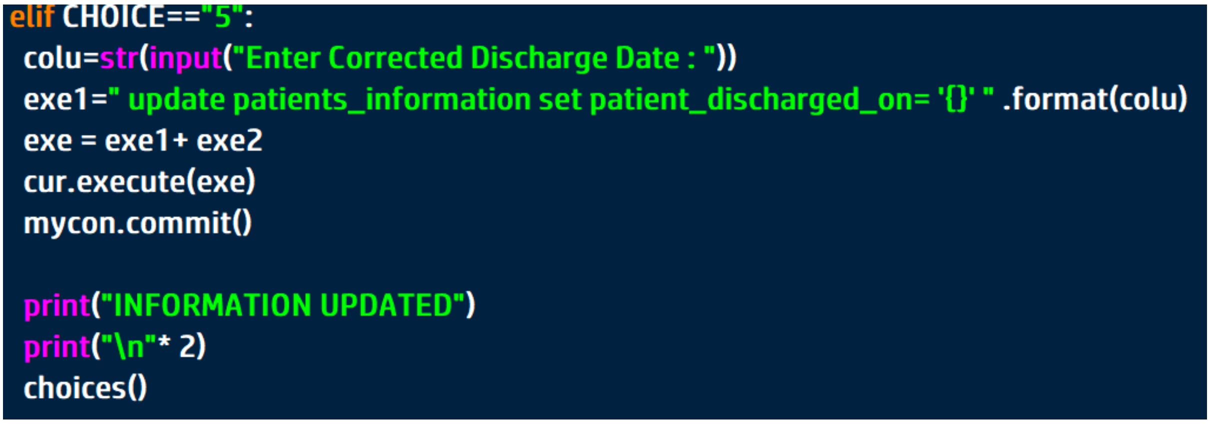


```
CHOICE=str(input( "Enter Choice Number (1/2/3/4/5/6) :"))

if CHOICE=="1":
  colu=str(input("Enter New Patient Id : "))
  exe1=" update patients_information set Patient_ID_number= '{}' " .format(colu)
  exe = exe1+ exe2
  cur.execute(exe)
  mycon.commit()

  print("INFORMATION UPDATED")
  print("\n"* 2)
  choices()

elif CHOICE=="2":
  colu=str(input("Enter Corrected : "))
  exe1=" update patients_information set name_of_patient= '{}' " .format(colu)
  exe = exe1+ exe2
  cur.execute(exe)
  mycon.commit()

  print("INFORMATION UPDATED")
  print("\n"* 2)
  choices()


elif CHOICE=="3":
```

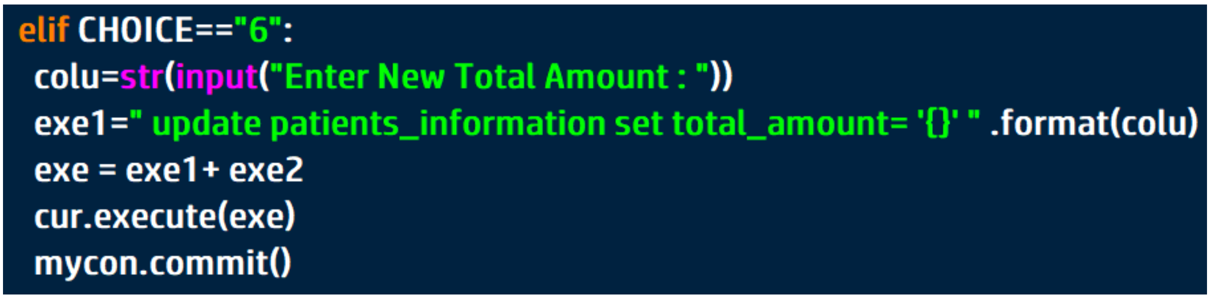
exe1 and exe2 have the queries in string form. So, when we do exe = exe1+exe2 then string concatenation takes place. The query is executed and changes are committed.



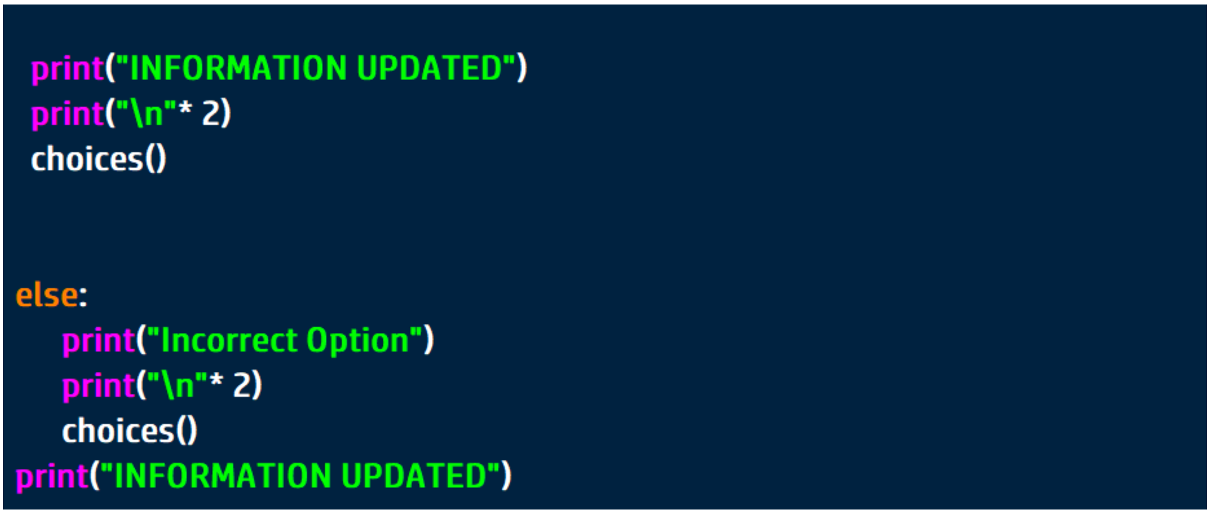
```
elif CHOICE=="3":
 colu=str(input("Enter New Symptoms : "))
 exe1=" update patients_information set disease= '{}' " .format(colu)
 exe = exe1+ exe2
 cur.execute(exe)
 mycon.commit()

 print("INFORMATION UPDATED")
 print("\n"* 2)
 choices()



elif CHOICE=="4":
 colu=str(input("Enter Corrected Admission Date : "))
 exe1=" update patients_information set patient_admitted_on= '{}' " .format(colu)
 exe = exe1+ exe2
 cur.execute(exe)
 mycon.commit()

 print("INFORMATION UPDATED")
 print("\n"* 2)
 choices()
```



```
elif CHOICE=="5":
 colu=str(input("Enter Corrected Discharge Date : "))
 exe1=" update patients_information set patient_discharged_on= '{}' " .format(colu)
 exe = exe1+ exe2
 cur.execute(exe)
 mycon.commit()

 print("INFORMATION UPDATED")
 print("\n"* 2)
 choices()
```

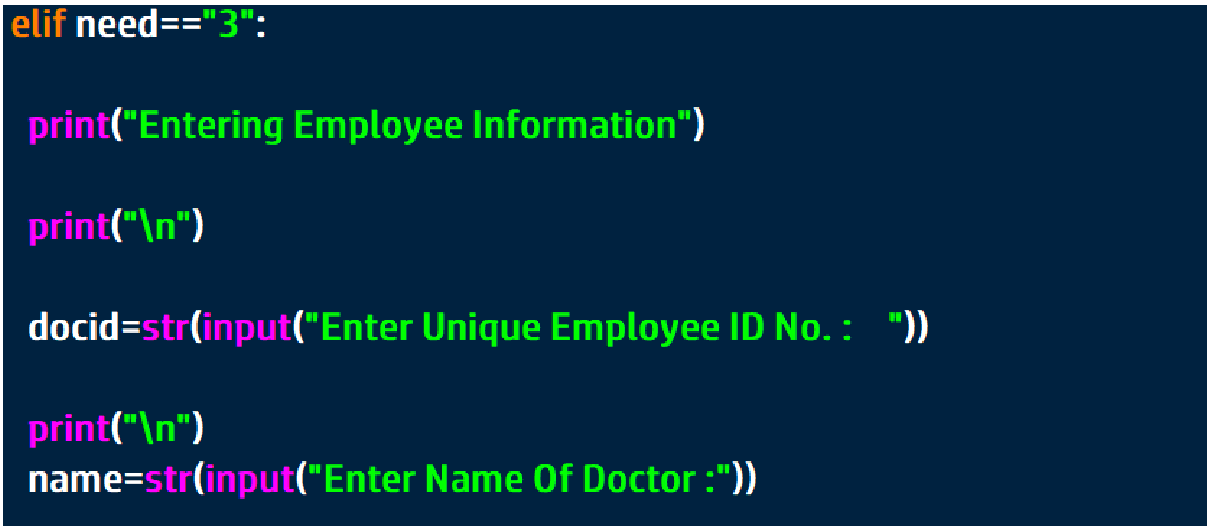


```
elif CHOICE=="6":
 colu=str(input("Enter New Total Amount : "))
 exe1=" update patients_information set total_amount= '{}' " .format(colu)
 exe = exe1+ exe2
 cur.execute(exe)
 mycon.commit()
```

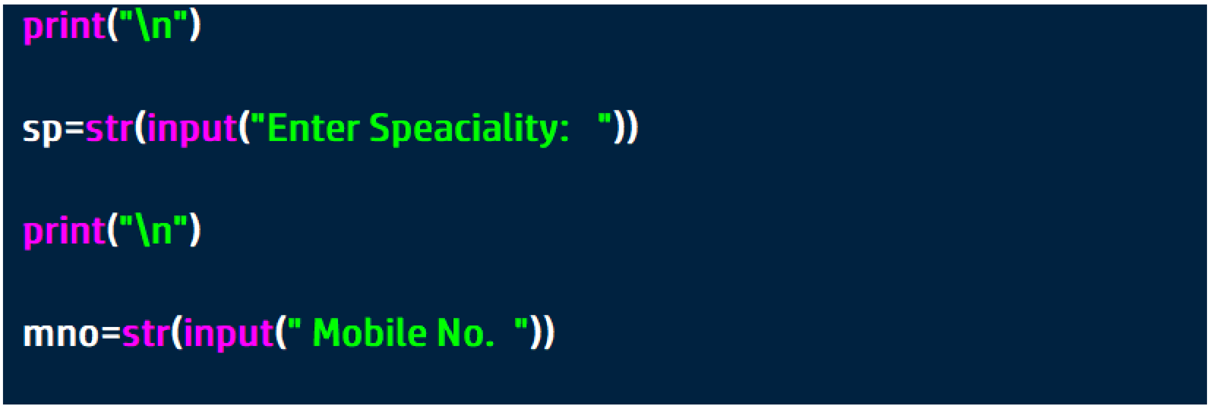


```
 print("INFORMATION UPDATED")
 print("\n"* 2)
 choices()


else:
    print("Incorrect Option")
    print("\n"* 2)
    choices()
print("INFORMATION UPDATED")
```

14.

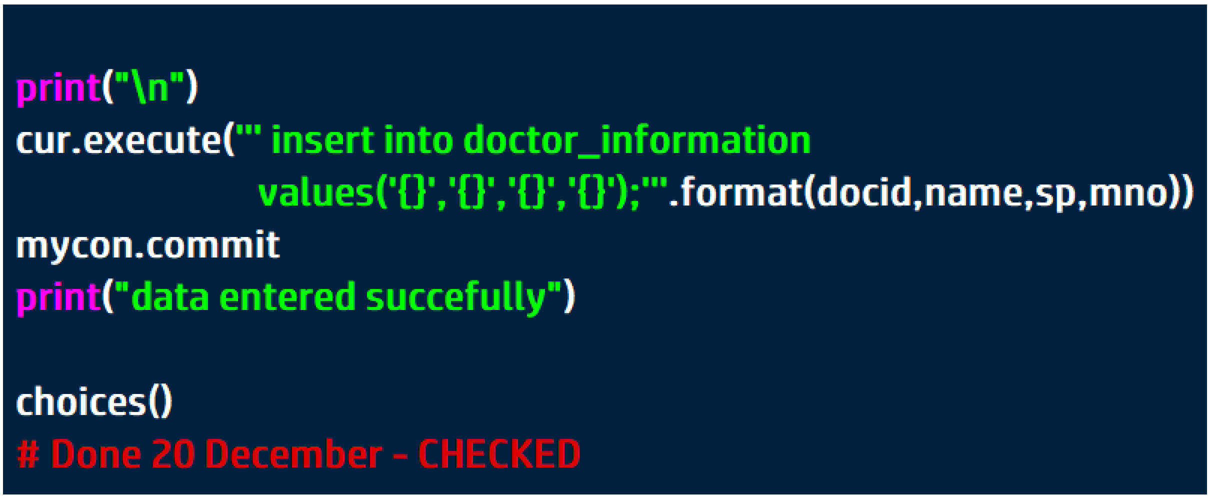


```
elif need=="3":

 print("Entering Employee Information")

 print("\n")

 docid=str(input("Enter Unique Employee ID No. :    "))

 print("\n")
 name=str(input("Enter Name Of Doctor :"))
```



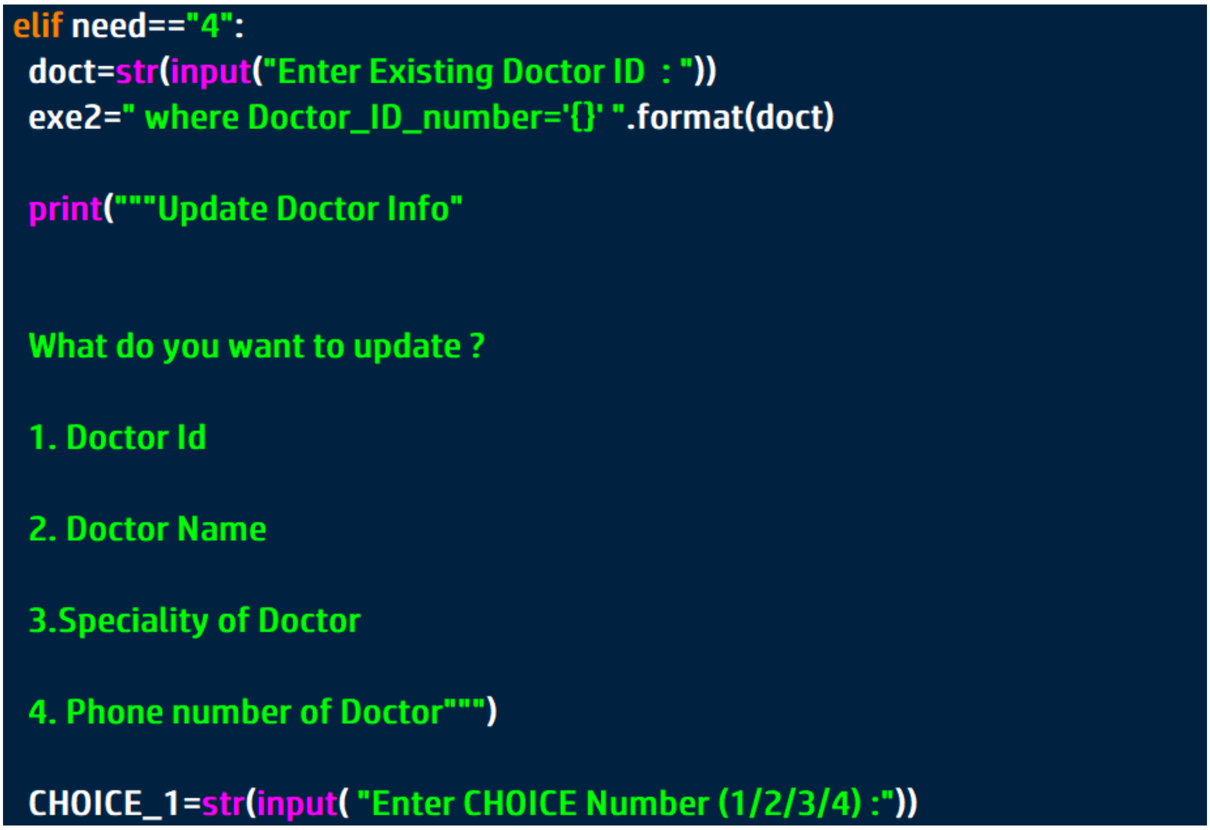
```
print("\n")

sp=str(input("Enter Speaciality:   "))

print("\n")

mno=str(input(" Mobile No.  "))
```

When the user wants to enter Data of a new doctor, this part of the code runs. Various details are inputted by the user and that is integrated to the SQL query using .format() method and query executed and changes are committed.

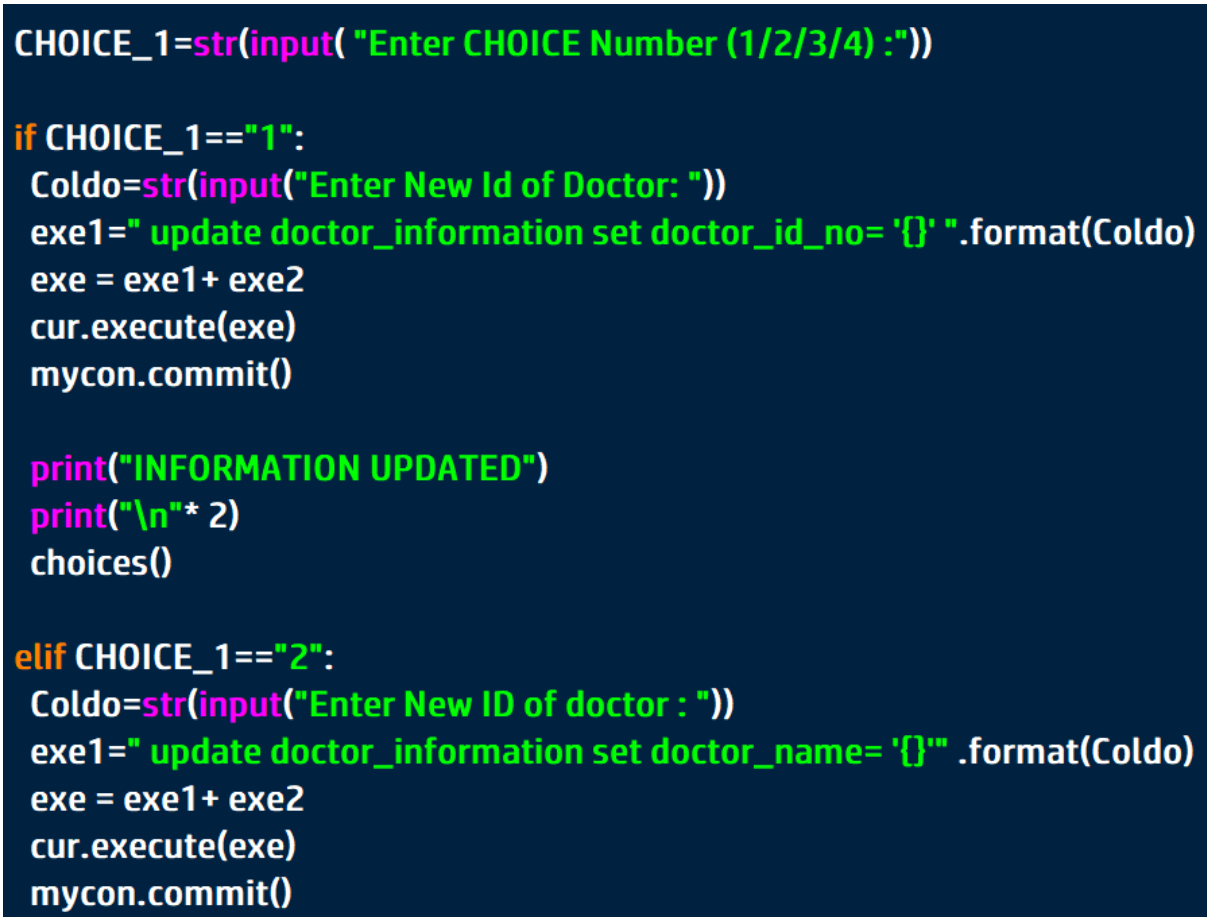


```
print("\n")
cur.execute(''' insert into doctor_information
                    values('{}','{}','{}','{}');'''.format(docid,name,sp,mno))
mycon.commit
print("data entered succefully")

choices()
# Done 20 December - CHECKED
```

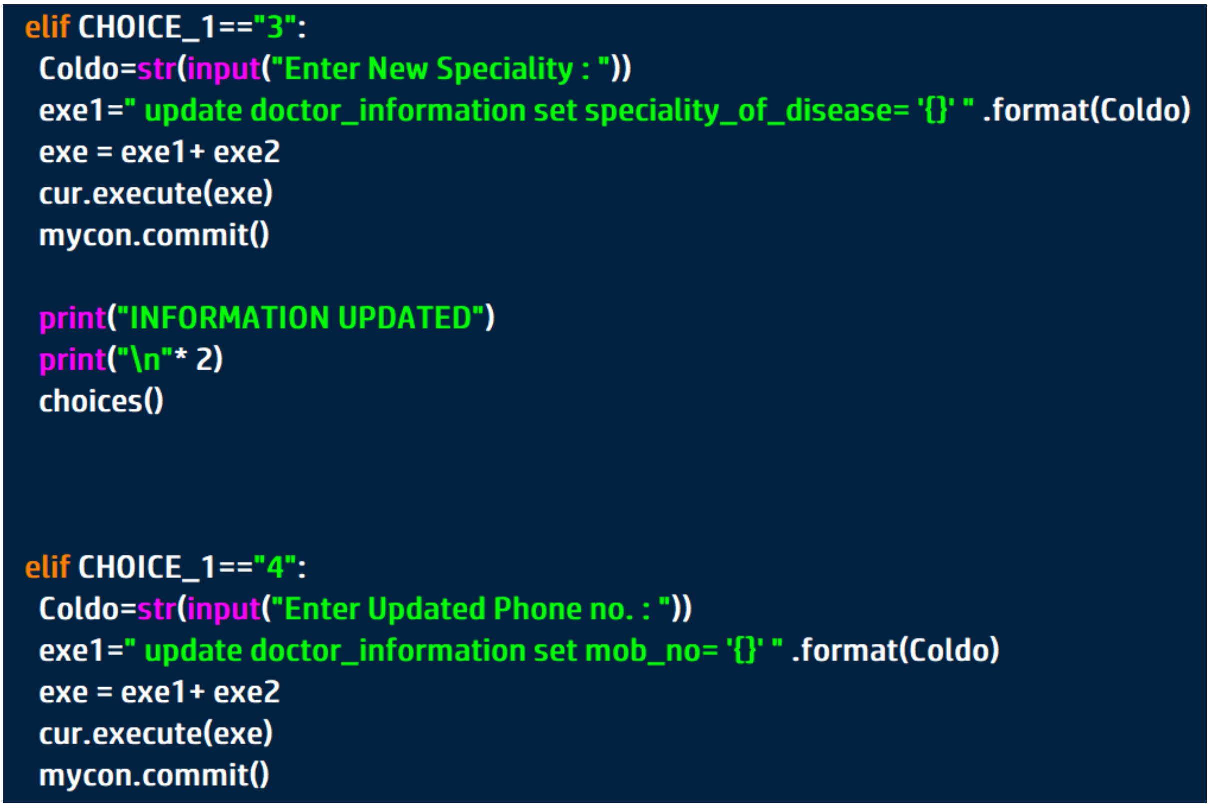
15.



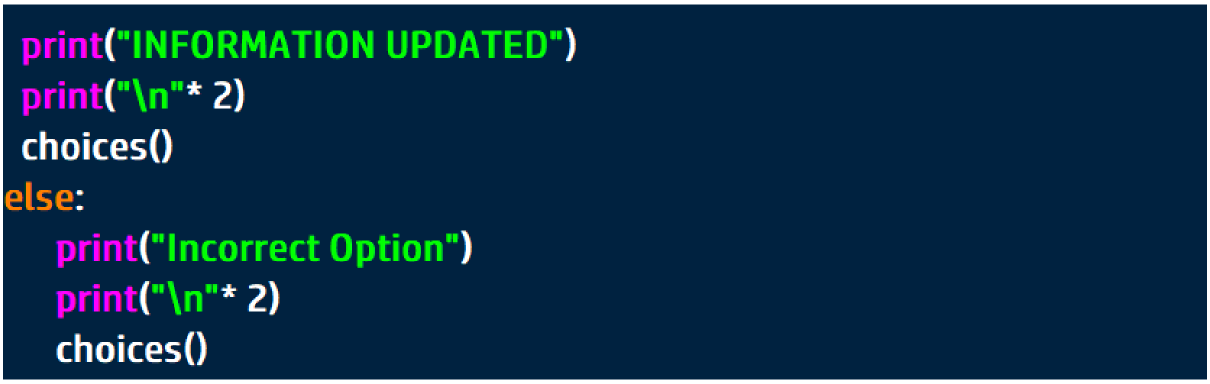
```
elif need=="4":
 doct=str(input("Enter Existing Doctor ID  : "))
 exe2=" where Doctor_ID_number='{}' ".format(doct)

 print("""Update Doctor Info"


 What do you want to update ?

 1. Doctor Id

 2. Doctor Name

 3.Speciality of Doctor

 4. Phone number of Doctor""")

 CHOICE_1=str(input( "Enter CHOICE Number (1/2/3/4) :"))
```



```
CHOICE_1=str(input( "Enter CHOICE Number (1/2/3/4) :"))

if CHOICE_1=="1":
 Coldo=str(input("Enter New Id of Doctor: "))
 exe1=" update doctor_information set doctor_id_no= '{}' ".format(Coldo)
 exe = exe1+ exe2
 cur.execute(exe)
 mycon.commit()

 print("INFORMATION UPDATED")
 print("\n"* 2)
 choices()

elif CHOICE_1=="2":
 Coldo=str(input("Enter New ID of doctor : "))
 exe1=" update doctor_information set doctor_name= '{}'" .format(Coldo)
 exe = exe1+ exe2
 cur.execute(exe)
 mycon.commit()
```



```
elif CHOICE_1=="3":
 Coldo=str(input("Enter New Speciality : "))
 exe1=" update doctor_information set speciality_of_disease= '{}' " .format(Coldo)
 exe = exe1+ exe2
 cur.execute(exe)
 mycon.commit()

 print("INFORMATION UPDATED")
 print("\n"* 2)
 choices()



elif CHOICE_1=="4":
 Coldo=str(input("Enter Updated Phone no. : "))
 exe1=" update doctor_information set mob_no= '{}' " .format(Coldo)
 exe = exe1+ exe2
 cur.execute(exe)
 mycon.commit()
```

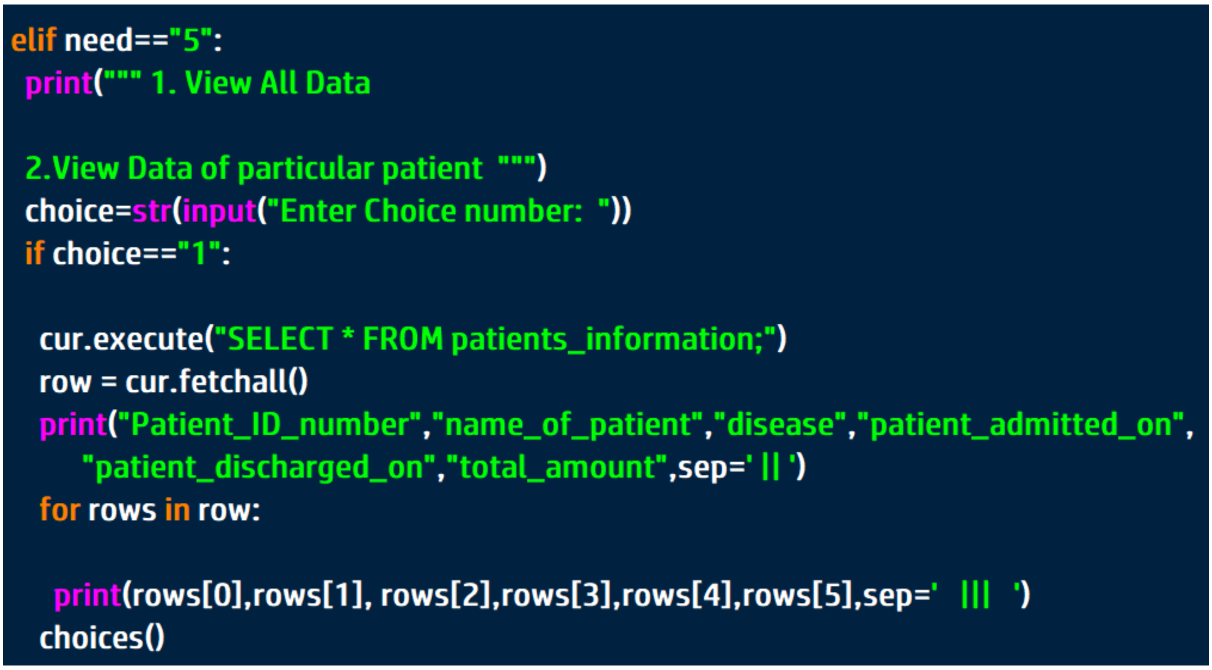


```
 print("INFORMATION UPDATED")
 print("\n"* 2)
 choices()
else:
    print("Incorrect Option")
    print("\n"* 2)
    choices()
```

16.

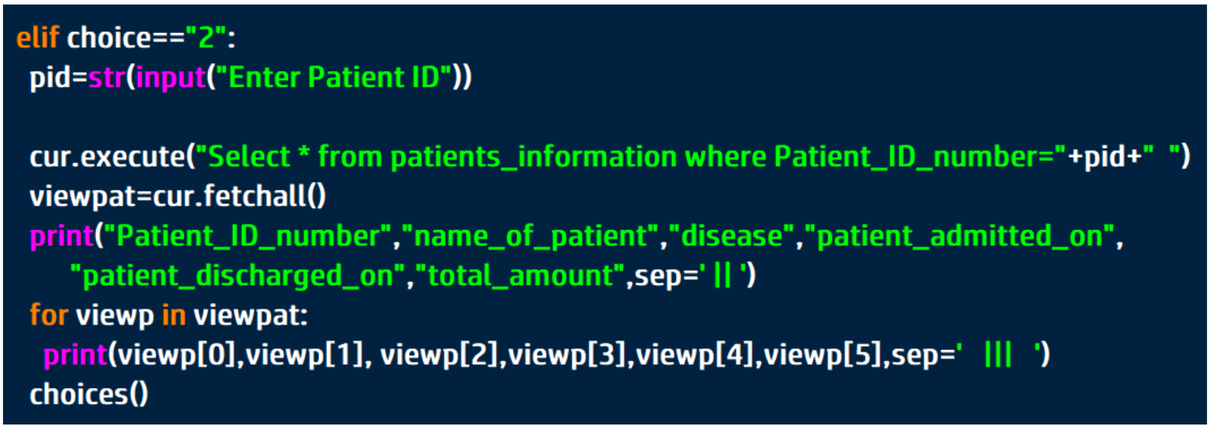
5th part of the else-if program is to view the data stored in MySQL Database. The user can view all the data of the table. **.fetchall()** fetches all the rows of a query result. It returns all the rows as a list of tuples. An empty list is returned if there is no record to fetch.

for rows in row: traverses the tuples fetched through fetchall() and prints them one by one.



```
elif need=="5":
 print(""" 1. View All Data

 2.View Data of particular patient  """)
 choice=str(input("Enter Choice number:  "))
 if choice=="1":

  cur.execute("SELECT * FROM patients_information;")
  row = cur.fetchall()
  print("Patient_ID_number","name_of_patient","disease","patient_admitted_on",
     "patient_discharged_on","total_amount",sep=' || ')
  for rows in row:

   print(rows[0],rows[1], rows[2],rows[3],rows[4],rows[5],sep='  |||  ')
  choices()
```

The **sep** parameter is primarily used to format the strings that need to be printed and add a separator between strings to be printed. Instead of using .format() method to integrate the data inputted by the user to the SQL query, we wrote a string inside two pluses inside the query string. This is also not in our syllabus and since this was a research project, we tried this.

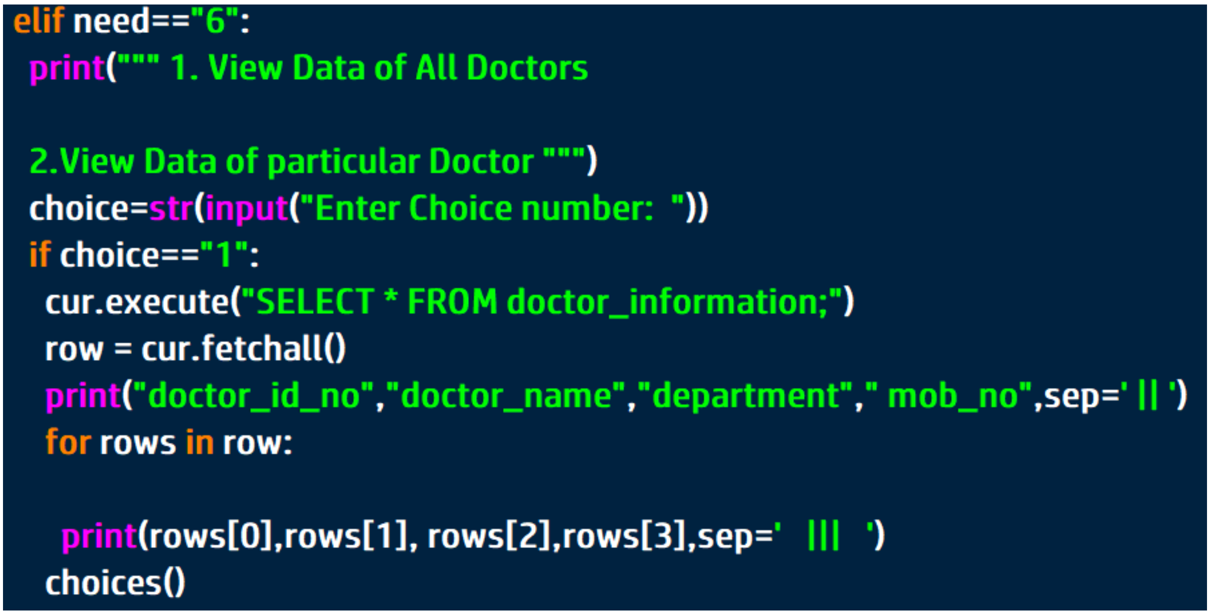


```
elif choice=="2":
  pid=str(input("Enter Patient ID"))

  cur.execute("Select * from patients_information where Patient_ID_number="+pid+"  ")
  viewpat=cur.fetchall()
  print("Patient_ID_number","name_of_patient","disease","patient_admitted_on",
      "patient_discharged_on","total_amount",sep=' || ')
  for viewp in viewpat:
   print(viewp[0],viewp[1], viewp[2],viewp[3],viewp[4],viewp[5],sep='   |||   ')
  choices()
```

17.

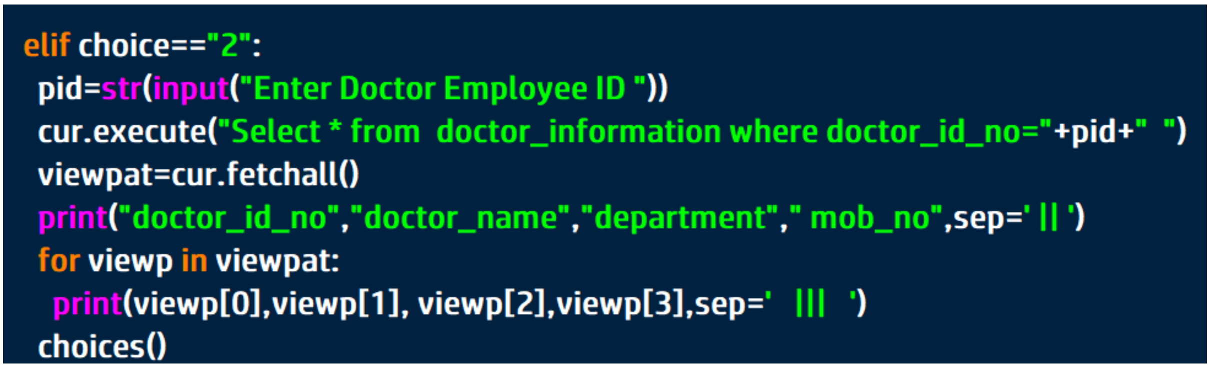
As similar to else-if code 5th, part of the code lets the user view data from the table doctor_information.

Either all data or data of a specific doctor can be seen by entering Doctor ID



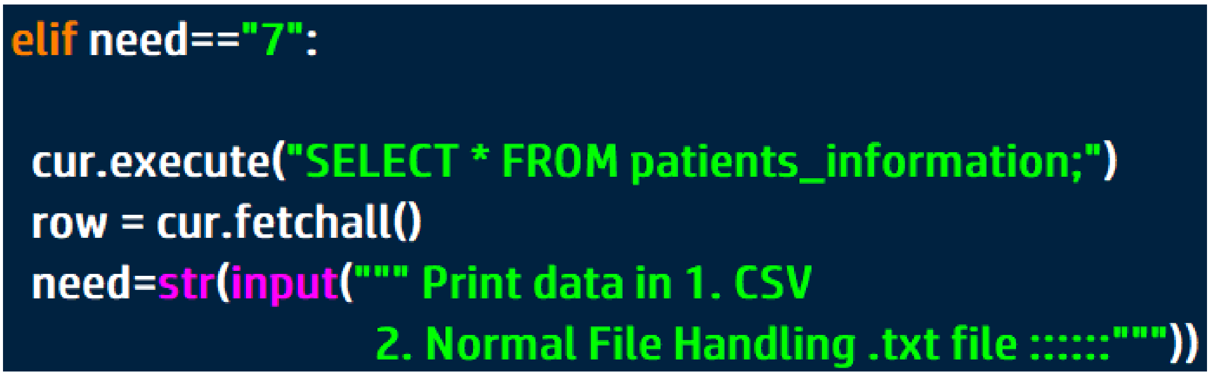
```
elif need=="6":
 print(""" 1. View Data of All Doctors

 2.View Data of particular Doctor """)
 choice=str(input("Enter Choice number:  "))
 if choice=="1":
  cur.execute("SELECT * FROM doctor_information;")
  row = cur.fetchall()
  print("doctor_id_no","doctor_name","department"," mob_no",sep=' || ')
  for rows in row:

   print(rows[0],rows[1], rows[2],rows[3],sep='   |||   ')
  choices()
```

.



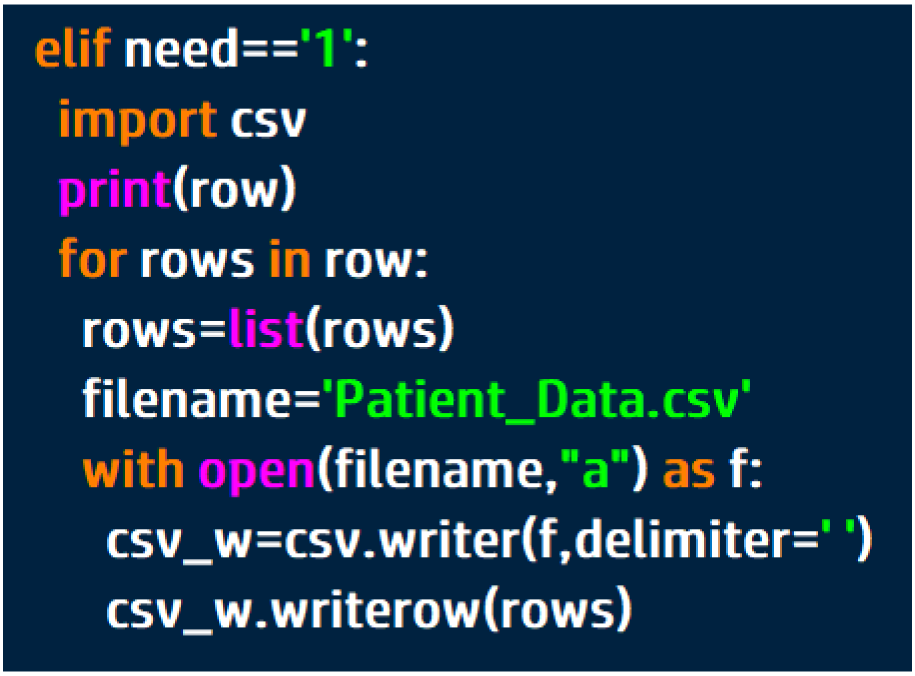
```
 elif choice=="2":
  pid=str(input("Enter Doctor Employee ID "))
  cur.execute("Select * from  doctor_information where doctor_id_no="+pid+"  ")
  viewpat=cur.fetchall()
  print("doctor_id_no","doctor_name","department"," mob_no",sep=' || ')
  for viewp in viewpat:
   print(viewp[0],viewp[1], viewp[2],viewp[3],sep='   |||   ')
  choices()
```

18.



```
elif need=="7":

 cur.execute("SELECT * FROM patients_information;")
 row = cur.fetchall()
 need=str(input(""" Print data in 1. CSV
                        2. Normal File Handling .txt file :::::"""))
```

Our Code lets the user get the data of doctors from the table *patients_information* in the form of .CSV or .txt



```
elif need=='1':
 import csv
 print(row)
 for rows in row:
  rows=list(rows)
  filename='Patient_Data.csv'
  with open(filename,"a") as f:
   csv_w=csv.writer(f,delimiter=' ')
   csv_w.writerow(rows)
```

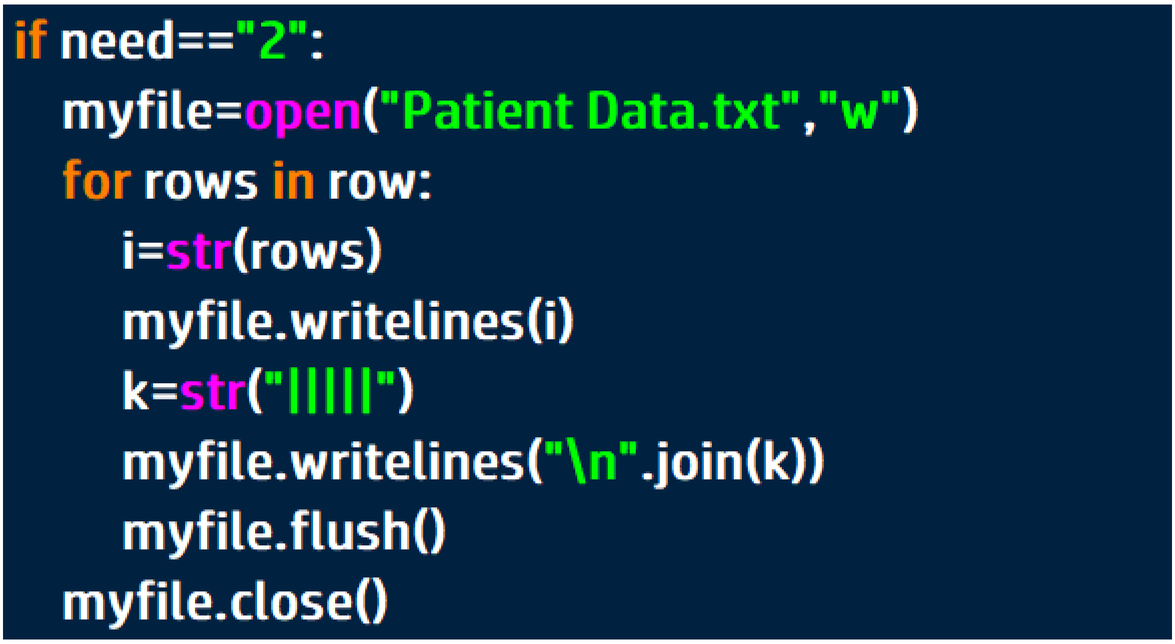
CSV Module has been used by us for this purpose.

open() function has been used to open a CSV a file and return file object.

csv.writer() function has been used to convert the data into delimited string. This string is written into CSV files using the **writerow()** function.

Writerow() allows us to write a list of fields to the file. We do not need to add a new line character to indicate the end of the line. It writes each row.

For loop has been used to by us to traverse the list of data stored in the variable row.



```
if need=="2":
    myfile=open("Patient Data.txt","w")
    for rows in row:
        i=str(rows)
        myfile.writelines(i)
        k=str("|||||")
        myfile.writelines("\n".join(k))
        myfile.flush()
    myfile.close()
```

Concepts of File Handling have been used by us to write the data from the table to a .TXT file.

We have used w mode to write the data into file.

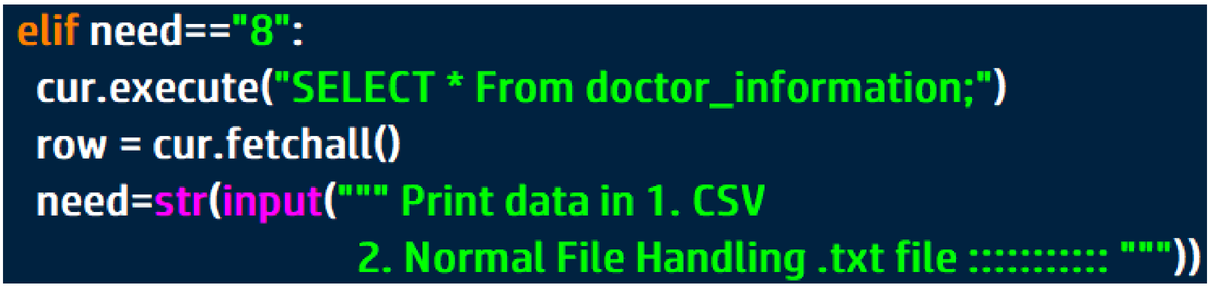
File objects are used to read and write data to a file. When we use open() function, python stores the reference of the mentioned file in the file-object.

<file handle>.writelines() writes all the strings in the list to the file referenced by *<file handle>.* We wrote in *“|n” .join(k)* in the code to so that it adds a new line after every string that is written to .txt file.

Flush() function has been used by us so that Python forces the writing of contents of buffer onto the storage.

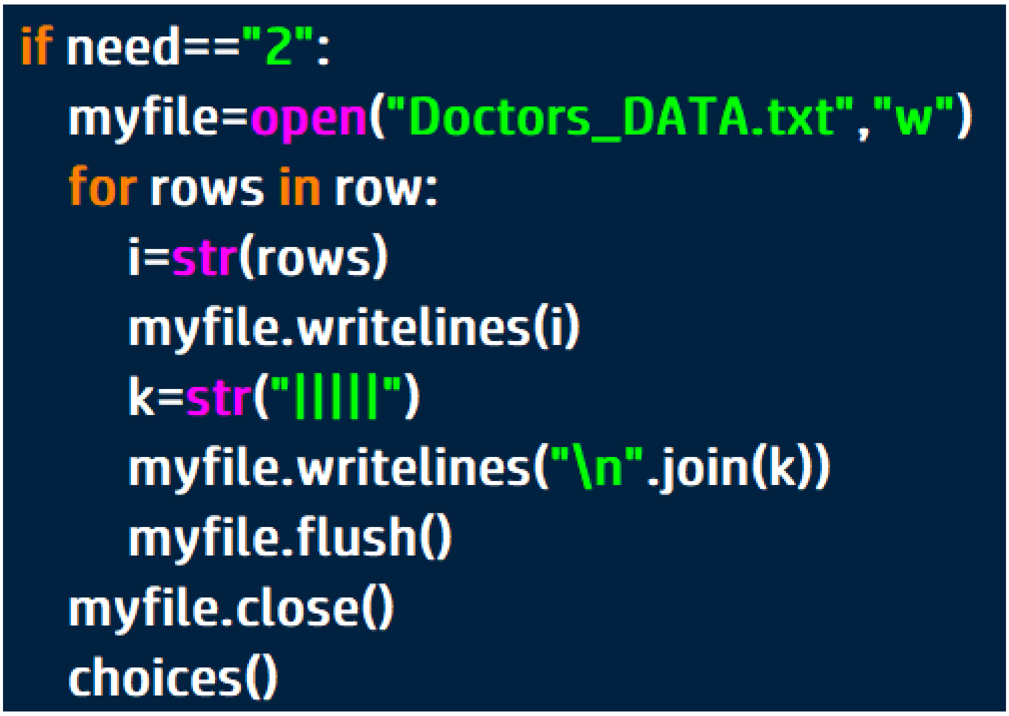
Close() function is used to break the link between file object and the file on the disk. After close(), no tasks can be performed on it. Precious data is not lost when we use it.

19.

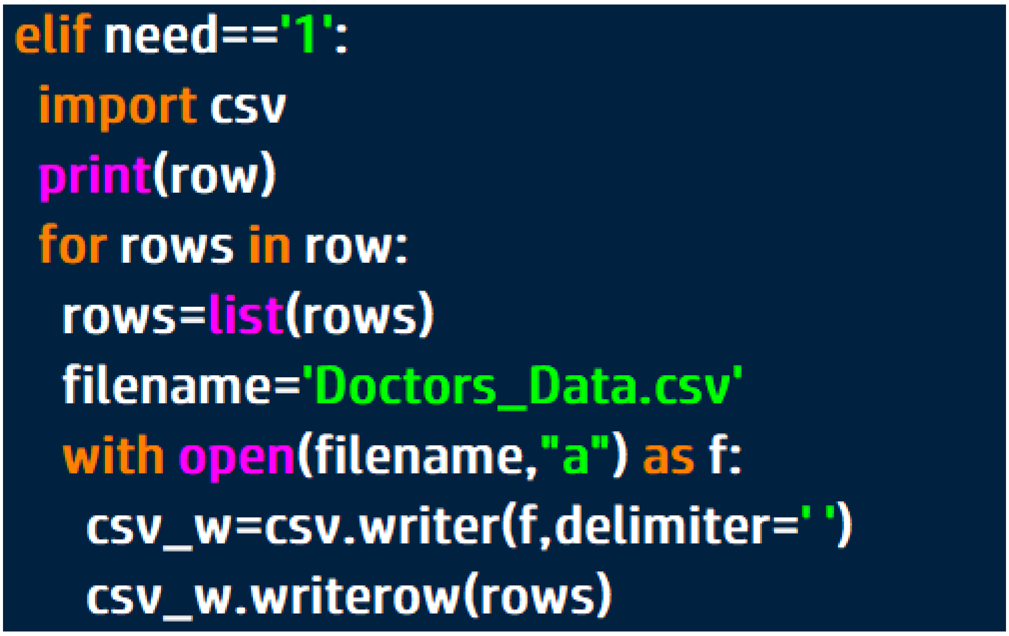


```
elif need=="8":
  cur.execute("SELECT * From doctor_information;")
  row = cur.fetchall()
  need=str(input(""" Print data in 1. CSV
                            2. Normal File Handling .txt file :::::::::: """))
```

Similar to the above code. The information of the doctor can also be taken by user in .CSV file and .txt file with this code.



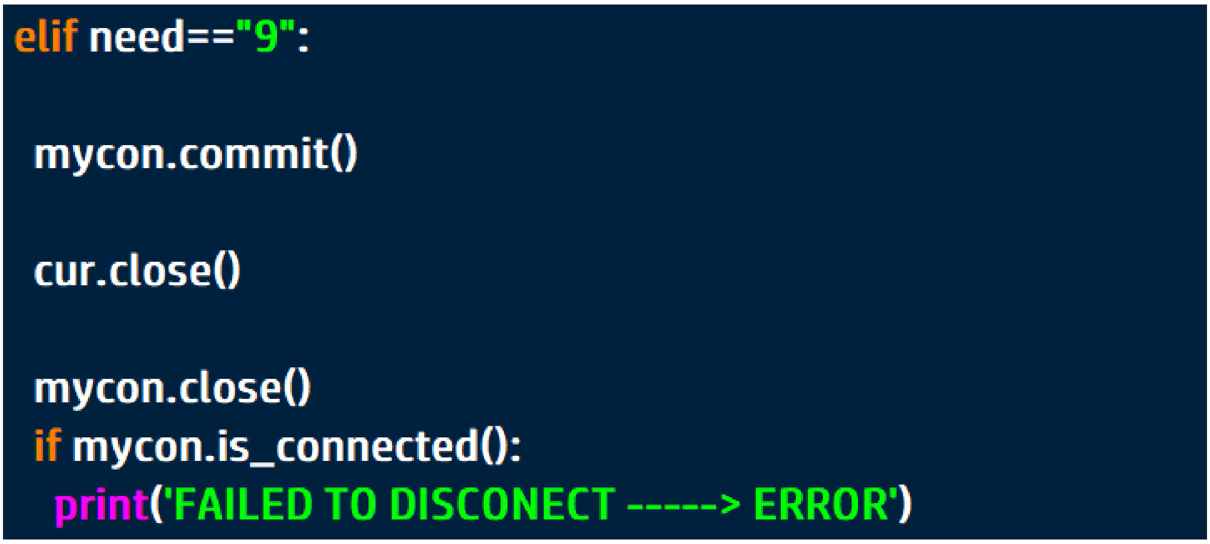
```
if need=="2":
    myfile=open("Doctors_DATA.txt","w")
    for rows in row:
        i=str(rows)
        myfile.writelines(i)
        k=str("|||||")
        myfile.writelines("\n".join(k))
        myfile.flush()
    myfile.close()
    choices()
```



```
elif need=='1':
 import csv
 print(row)
 for rows in row:
  rows=list(rows)
  filename='Doctors_Data.csv'
  with open(filename,"a") as f:
   csv_w=csv.writer(f,delimiter=' ')
   csv_w.writerow(rows)
```

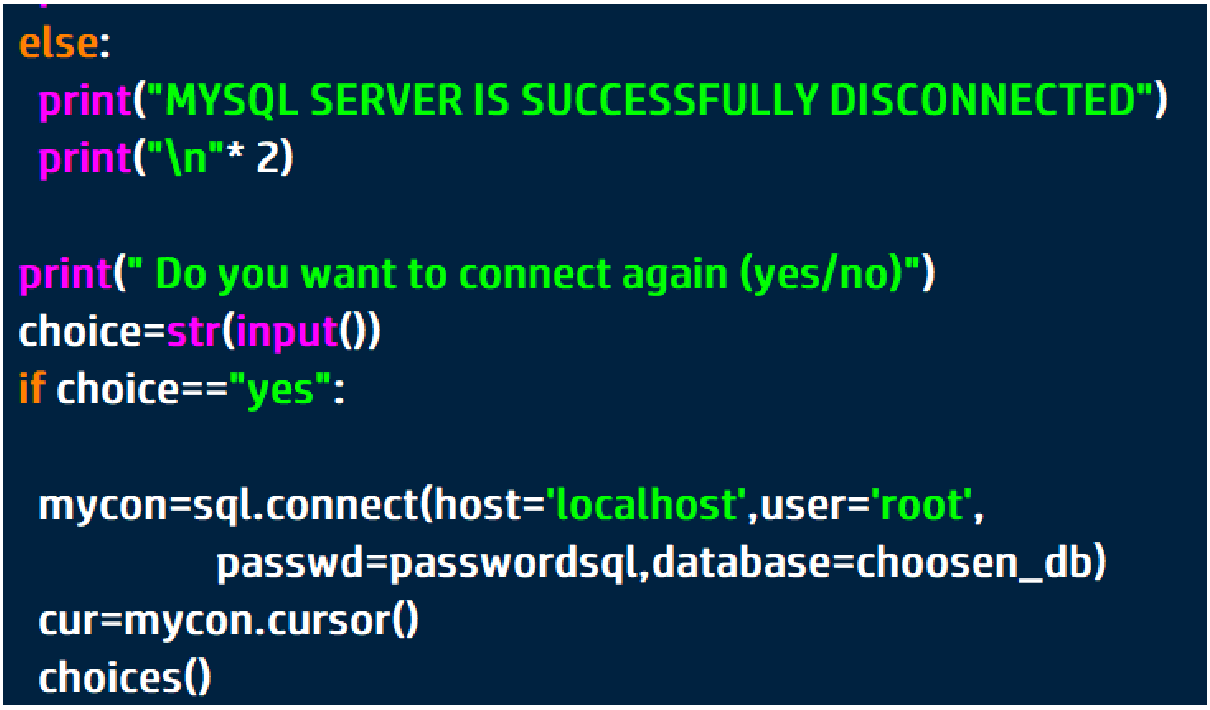
20.

The user can close the connection between MySQL and Python with the help of the following code –



```
elif need=="9":

 mycon.commit()

 cur.close()

 mycon.close()
 if mycon.is_connected():
  print('FAILED TO DISCONECT -----> ERROR')
```

It also print error if it is unable to disconnect.

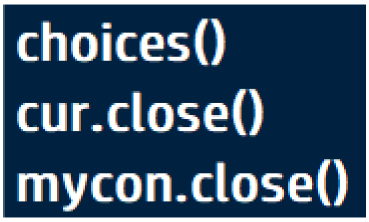


```
else:
  print("MYSQL SERVER IS SUCCESSFULLY DISCONNECTED")
  print("\n"* 2)

print(" Do you want to connect again (yes/no)")
choice=str(input())
if choice=="yes":

  mycon=sql.connect(host='localhost',user='root',
                  passwd=passwordsql,database=choosen_db)
  cur=mycon.cursor()
  choices()
```

The user can initiate the connection again if it needs to be.

21.

As told earlier, our whole program (main program) is defined under a single function. So at the end of the program we have called that function.

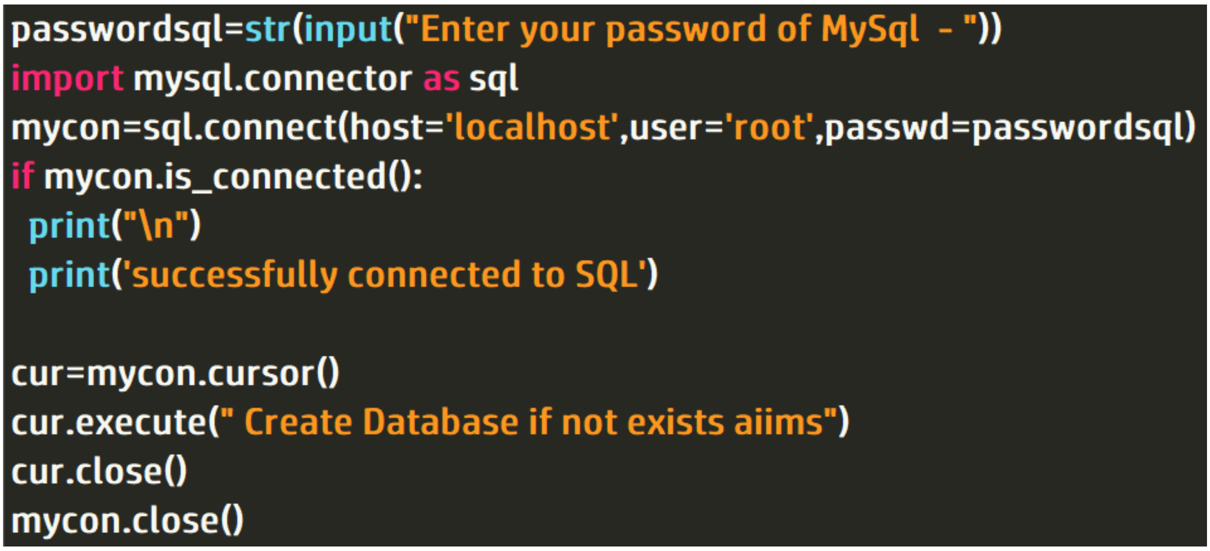


```
choices()
cur.close()
mycon.close()
```

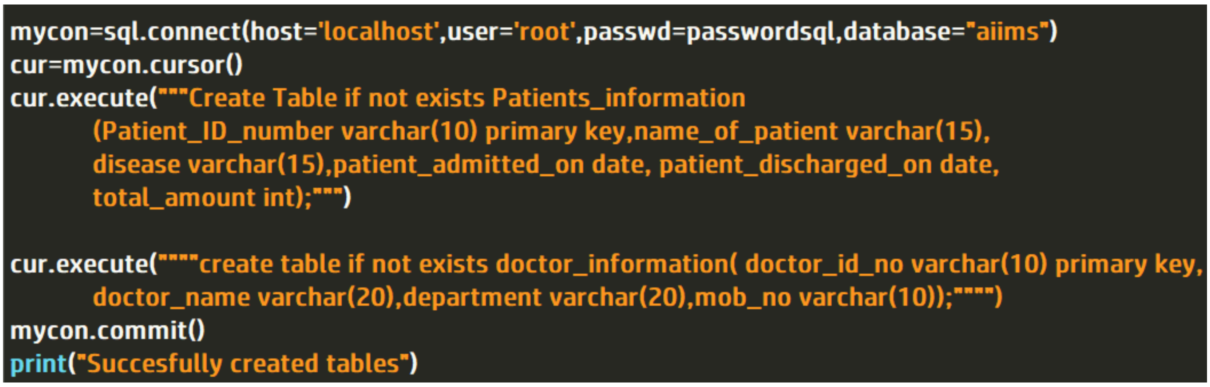
ALGORITHM – 2

(For database Creation)

We have also created a 2nd program that creates the required database to run/test the 1st program in any computer.

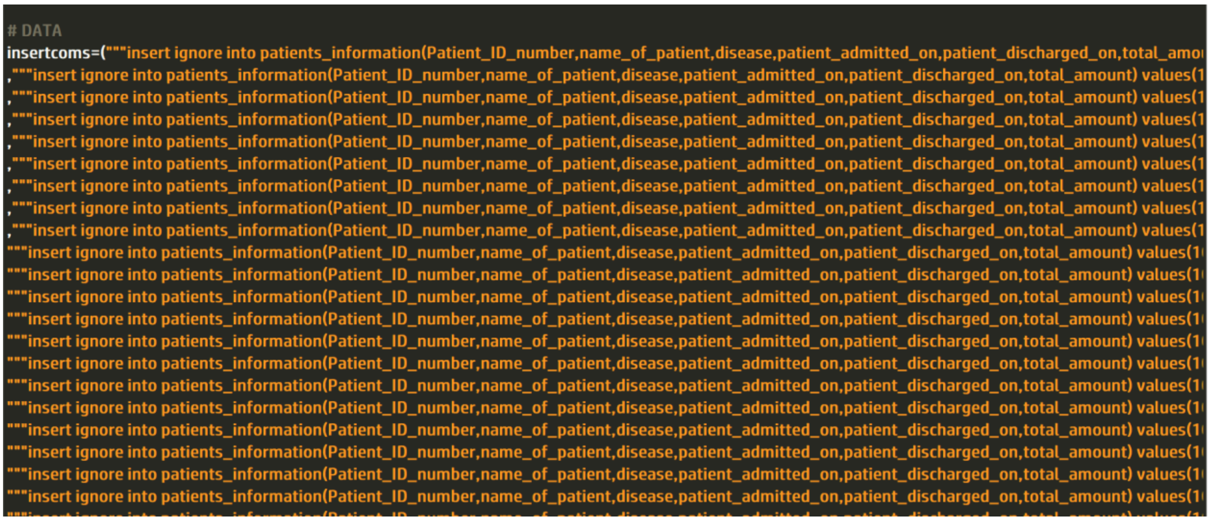


```
passwordsql=str(input("Enter your password of MySql  - "))
import mysql.connector as sql
mycon=sql.connect(host='localhost',user='root',passwd=passwordsql)
if mycon.is_connected():
  print("\n")
  print('successfully connected to SQL')

cur=mycon.cursor()
cur.execute(" Create Database if not exists aiims")
cur.close()
mycon.close()
```

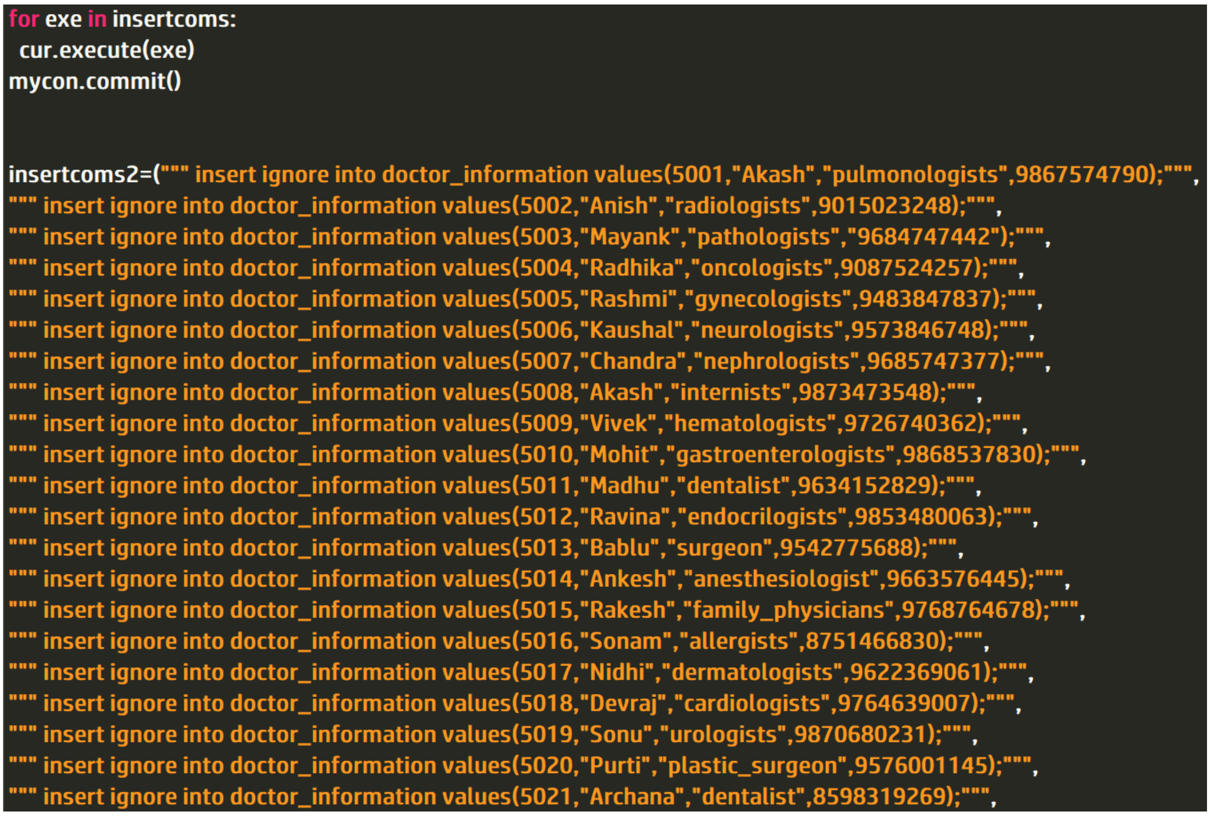


```
mycon=sql.connect(host='localhost',user='root',passwd=passwordsql,database="aiims")
cur=mycon.cursor()
cur.execute("""Create Table if not exists Patients_information
        (Patient_ID_number varchar(10) primary key,name_of_patient varchar(15),
        disease varchar(15),patient_admitted_on date, patient_discharged_on date,
        total_amount int);""")

cur.execute(""""create table if not exists doctor_information( doctor_id_no varchar(10) primary key,
        doctor_name varchar(20),department varchar(20),mob_no varchar(10));"""")
mycon.commit()
print("Succesfully created tables")
```

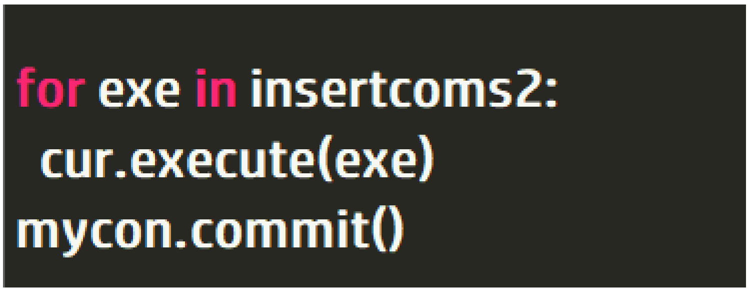
We also found a way so that if the data already exists then it will not give error, we have included *ignore into* with every statement.



```
# DATA
insertcoms=("""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amo
,"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
,"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
,"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
,"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
,"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
,"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
,"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
,"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
"""insert ignore into patients_information(Patient_ID_number,name_of_patient,disease,patient_admitted_on,patient_discharged_on,total_amount) values(1
```

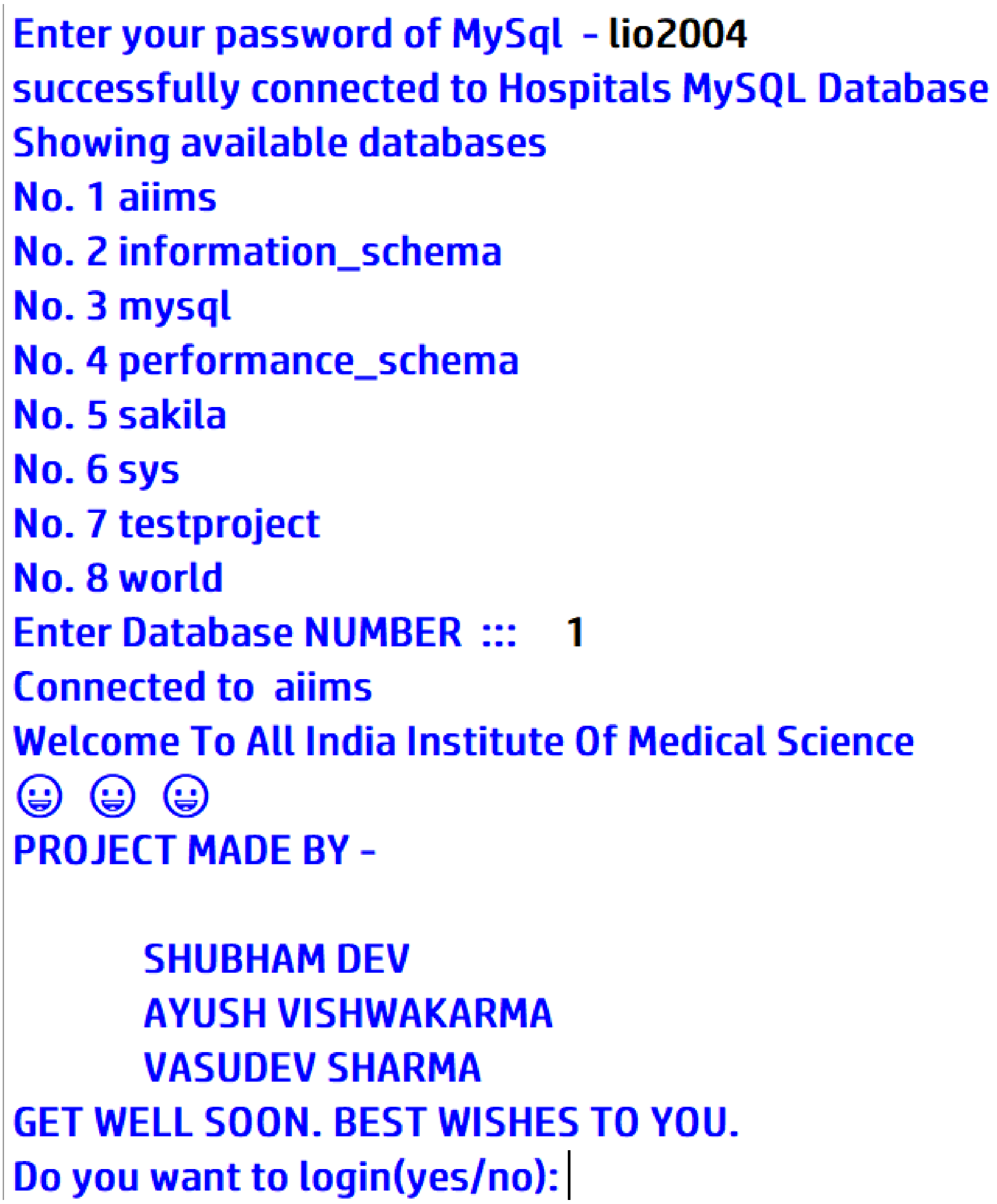


```
for exe in insertcoms:
 cur.execute(exe)
mycon.commit()


insertcoms2=(""" insert ignore into doctor_information values(5001,"Akash","pulmonologists",9867574790);""",
""" insert ignore into doctor_information values(5002,"Anish","radiologists",9015023248);""",
""" insert ignore into doctor_information values(5003,"Mayank","pathologists","9684747442");""",
""" insert ignore into doctor_information values(5004,"Radhika","oncologists",9087524257);""",
""" insert ignore into doctor_information values(5005,"Rashmi","gynecologists",9483847837);""",
""" insert ignore into doctor_information values(5006,"Kaushal","neurologists",9573846748);""",
""" insert ignore into doctor_information values(5007,"Chandra","nephrologists",9685747377);""",
""" insert ignore into doctor_information values(5008,"Akash","internists",9873473548);""",
""" insert ignore into doctor_information values(5009,"Vivek","hematologists",9726740362);""",
""" insert ignore into doctor_information values(5010,"Mohit","gastroenterologists",9868537830);""",
""" insert ignore into doctor_information values(5011,"Madhu","dentalist",9634152829);""",
""" insert ignore into doctor_information values(5012,"Ravina","endocrilogists",9853480063);""",
""" insert ignore into doctor_information values(5013,"Bablu","surgeon",9542775688);""",
""" insert ignore into doctor_information values(5014,"Ankesh","anesthesiologist",9663576445);""",
""" insert ignore into doctor_information values(5015,"Rakesh","family_physicians",9768764678);""",
""" insert ignore into doctor_information values(5016,"Sonam","allergists",8751466830);""",
""" insert ignore into doctor_information values(5017,"Nidhi","dermatologists",9622369061);""",
""" insert ignore into doctor_information values(5018,"Devraj","cardiologists",9764639007);""",
""" insert ignore into doctor_information values(5019,"Sonu","urologists",9870680231);""",
""" insert ignore into doctor_information values(5020,"Purti","plastic_surgeon",9576001145);""",
""" insert ignore into doctor_information values(5021,"Archana","dentalist",8598319269);""",
```

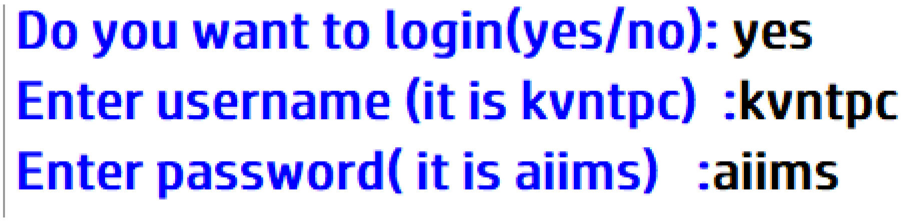


```
for exe in insertcoms2:
 cur.execute(exe)
mycon.commit()
```

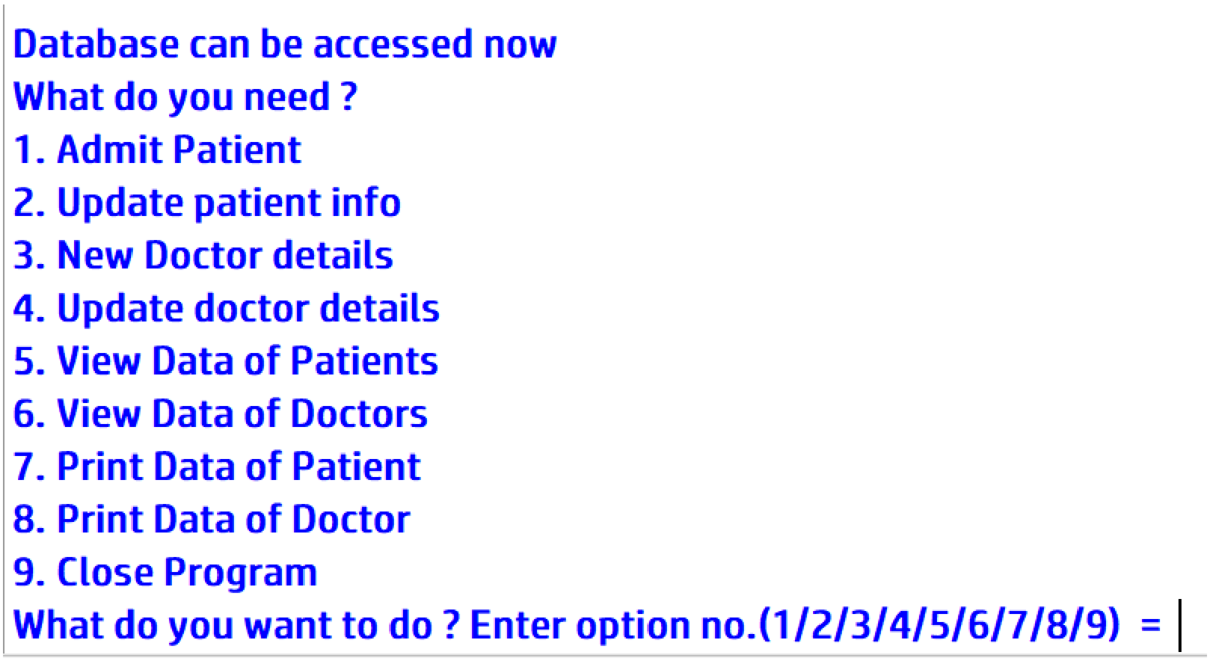
OUTPUT



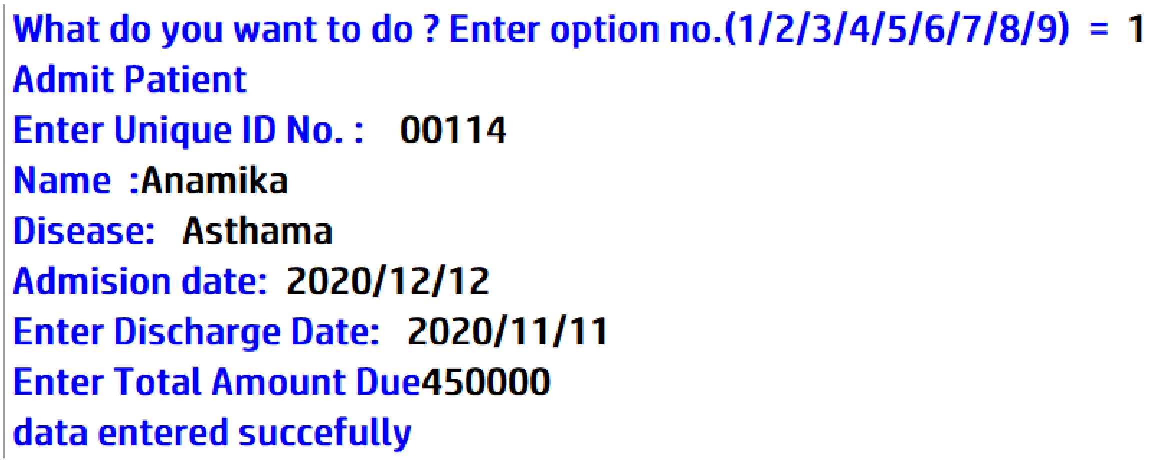
```
Enter your password of MySql  - lio2004
successfully connected to Hospitals MySQL Database
Showing available databases
No. 1 aiims
No. 2 information_schema
No. 3 mysql
No. 4 performance_schema
No. 5 sakila
No. 6 sys
No. 7 testproject
No. 8 world
Enter Database NUMBER  :::    1
Connected to  aiims
Welcome To All India Institute Of Medical Science
😀  😀  😀
PROJECT MADE BY -

            SHUBHAM DEV
            AYUSH VISHWAKARMA
            VASUDEV SHARMA
GET WELL SOON. BEST WISHES TO YOU.
Do you want to login(yes/no):
```



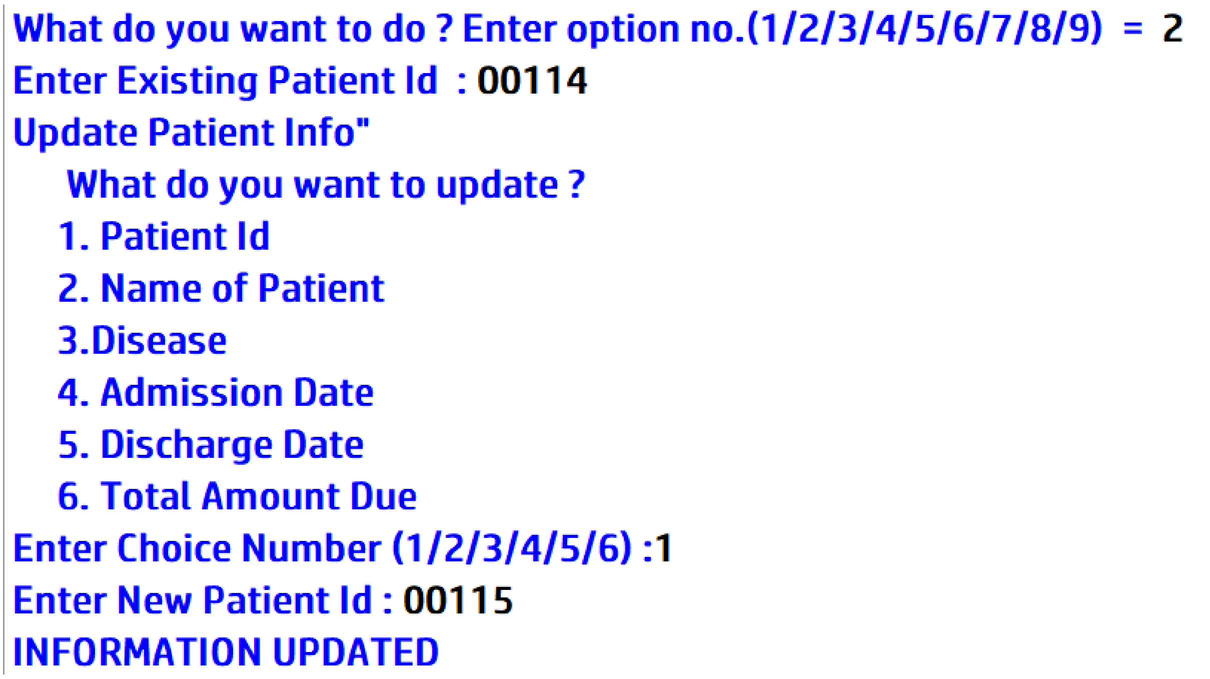
```
Do you want to login(yes/no): yes
Enter username (it is kvntpc)  :kvntpc
Enter password( it is aiims)   :aiims
```



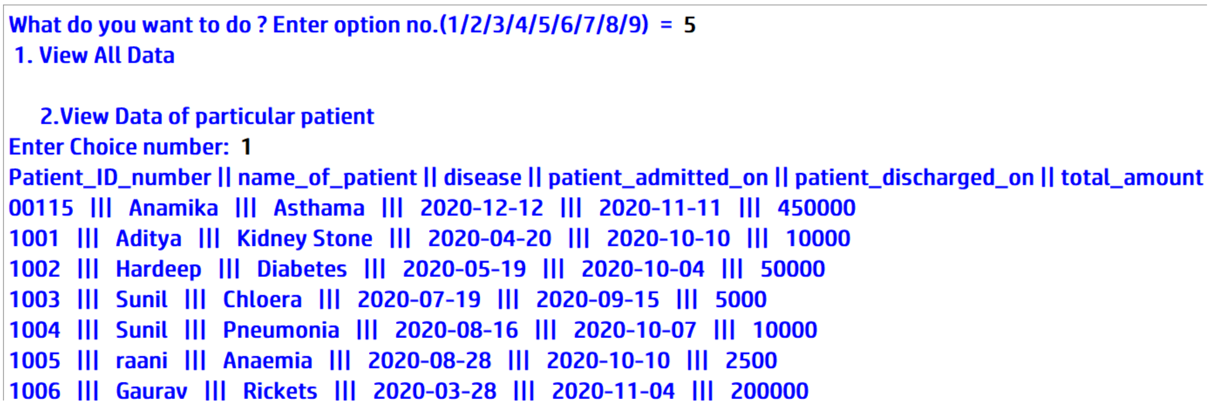
```
Database can be accessed now
What do you need ?
1. Admit Patient
2. Update patient info
3. New Doctor details
4. Update doctor details
5. View Data of Patients
6. View Data of Doctors
7. Print Data of Patient
8. Print Data of Doctor
9. Close Program
What do you want to do ? Enter option no.(1/2/3/4/5/6/7/8/9)  =
```



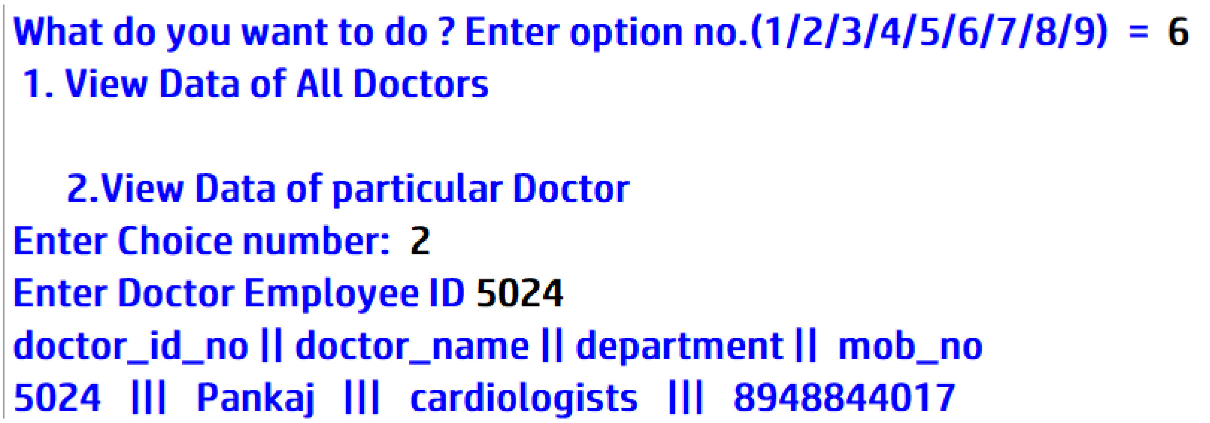
```
What do you want to do ? Enter option no.(1/2/3/4/5/6/7/8/9)  =  1
Admit Patient
Enter Unique ID No. :   00114
Name  :Anamika
Disease:   Asthama
Admision date:  2020/12/12
Enter Discharge Date:   2020/11/11
Enter Total Amount Due450000
data entered succefully
```



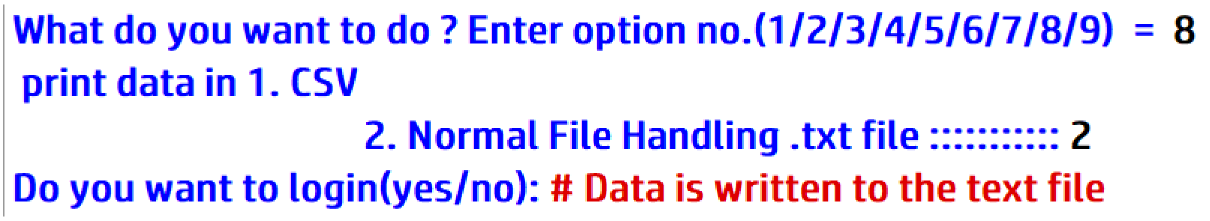
```
What do you want to do ? Enter option no.(1/2/3/4/5/6/7/8/9)  =  2
Enter Existing Patient Id  : 00114
Update Patient Info"
    What do you want to update ?
   1. Patient Id
   2. Name of Patient
   3.Disease
   4. Admission Date
   5. Discharge Date
   6. Total Amount Due
Enter Choice Number (1/2/3/4/5/6) :1
Enter New Patient Id : 00115
INFORMATION UPDATED
```



```
What do you want to do ? Enter option no.(1/2/3/4/5/6/7/8/9)  =  5
 1. View All Data

     2.View Data of particular patient
Enter Choice number:  1
Patient_ID_number || name_of_patient || disease || patient_admitted_on || patient_discharged_on || total_amount
00115  |||  Anamika  |||  Asthama  |||  2020-12-12  |||  2020-11-11  |||  450000
1001  |||  Aditya  |||  Kidney Stone  |||  2020-04-20  |||  2020-10-10  |||  10000
1002  |||  Hardeep  |||  Diabetes  |||  2020-05-19  |||  2020-10-04  |||  50000
1003  |||  Sunil  |||  Chloera  |||  2020-07-19  |||  2020-09-15  |||  5000
1004  |||  Sunil  |||  Pneumonia  |||  2020-08-16  |||  2020-10-07  |||  10000
1005  |||  raani  |||  Anaemia  |||  2020-08-28  |||  2020-10-10  |||  2500
1006  |||  Gaurav  |||  Rickets  |||  2020-03-28  |||  2020-11-04  |||  200000
```

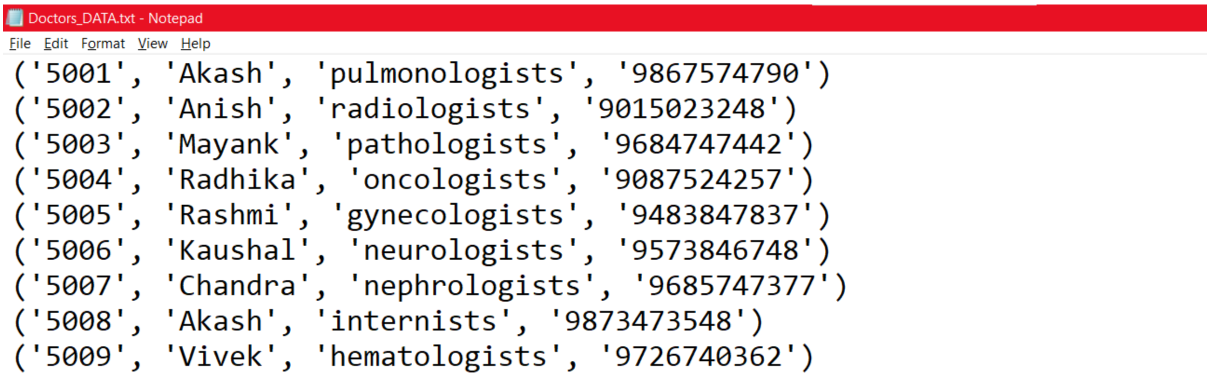


```
What do you want to do ? Enter option no.(1/2/3/4/5/6/7/8/9)  =  6
 1. View Data of All Doctors


     2.View Data of particular Doctor
Enter Choice number:  2
Enter Doctor Employee ID 5024
doctor_id_no || doctor_name || department ||  mob_no
5024  |||  Pankaj  |||  cardiologists  |||  8948844017
```

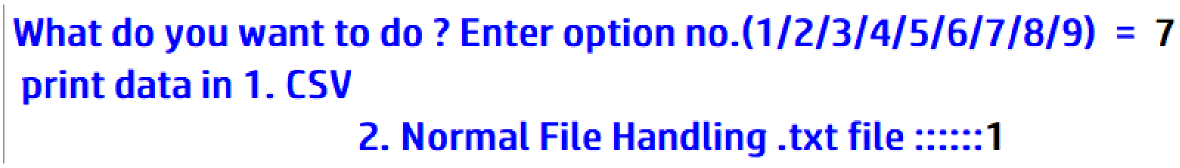


```
What do you want to do ? Enter option no.(1/2/3/4/5/6/7/8/9)  =  8
 print data in 1. CSV
                           2. Normal File Handling .txt file ::::::::::: 2
Do you want to login(yes/no): # Data is written to the text file
```

Output of the text file written by the code is shown below:

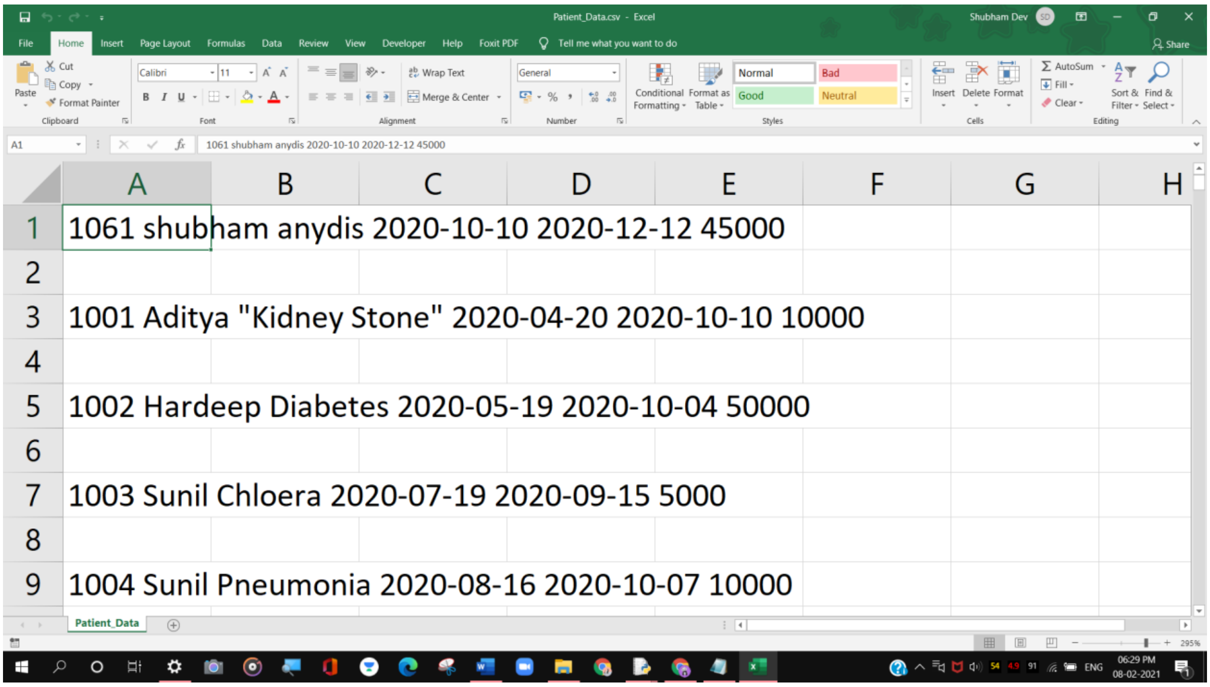


```
('5001', 'Akash', 'pulmonologists', '9867574790')
('5002', 'Anish', 'radiologists', '9015023248')
('5003', 'Mayank', 'pathologists', '9684747442')
('5004', 'Radhika', 'oncologists', '9087524257')
('5005', 'Rashmi', 'gynecologists', '9483847837')
('5006', 'Kaushal', 'neurologists', '9573846748')
('5007', 'Chandra', 'nephrologists', '9685747377')
('5008', 'Akash', 'internists', '9873473548')
('5009', 'Vivek', 'hematologists', '9726740362')
```



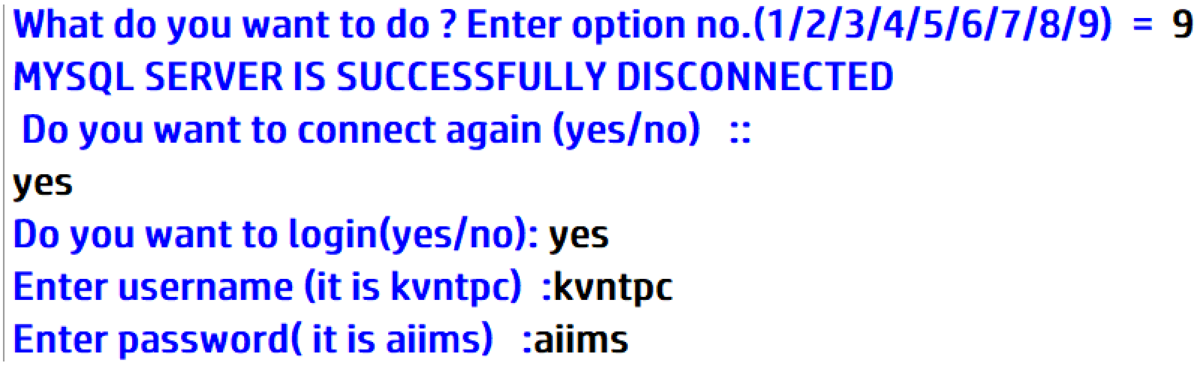
```
What do you want to do ? Enter option no.(1/2/3/4/5/6/7/8/9)  =  7
 print data in 1. CSV
                                 2. Normal File Handling .txt file ::::::1
```

Output of the CSV file written by the code is shown below:



```
A1    fx  1061 shubham anydis 2020-10-10 2020-12-12 45000

1  1061 shubham anydis 2020-10-10 2020-12-12 45000
2
3  1001 Aditya "Kidney Stone" 2020-04-20 2020-10-10 10000
4
5  1002 Hardeep Diabetes 2020-05-19 2020-10-04 50000
6
7  1003 Sunil Chloera 2020-07-19 2020-09-15 5000
8
9  1004 Sunil Pneumonia 2020-08-16 2020-10-07 10000
```

The connection is terminated and is shown in the following screenshot :



```
What do you want to do ? Enter option no.(1/2/3/4/5/6/7/8/9)  =  9
MYSQL SERVER IS SUCCESSFULLY DISCONNECTED
 Do you want to connect again (yes/no)   ::
yes
Do you want to login(yes/no): yes
Enter username (it is kvntpc)  :kvntpc
Enter password( it is aiims)   :aiims
```

REFERENCES

Books

- Computer Science with Python – Sumita Arora (Class XI)
- Computer Science with Python – Sumita Arora (Class XII)
- Computer Science with Python - Preeti Arora (Class XII)

Websites

- https://www.w3schools.com/
- https://www.codecademy.com/
- https://developers.google.com/
- https://www.learnpython.org/
- https://www.learncbse.in/