ADVANCED PROGRAMMING + DBMS

MID-SEMESTER ASSESSMENT CARD

TEAM 15

AMRITA VISHWA VIDYAPEETHAM, CHENNAI

TEAM MEMBERS

ANIKET MISHRA
CH.EN.U4CYS20004

JYOTHIKA NAMBIAR
CH.EN.U4CYS20032

KADIYALA RISHIKESH
CH.EN.U4CYS20033

SHRADDHA CHOPRA
CH.EN.U4CYS20068

SUBMITTED TO

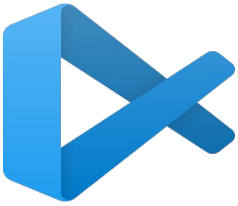
DR. V MURALIDARAN
Assistant Professor
Department of Computer Science and Engineering

DR. RAGUPATHY P
Assistant Professor
Department of Computer Science and Engineering

ADVANCED PROGRAMMING

DATABASE MANAGEMENT SYSTEMS

USING THE SOFTWARE



SUBMITED ON 18TH DECEMBER, 2021

BY **TEAM 15**
CYBER SECURITY
ODD SEMESTER
2ND YEAR

MID-SEMESTER ASSESSMENT CARD

ADVANCED PROGRAMMING + DATABASE MANAGEMENT SYSTEMS

TABLE CHOSEN:

The name of our project is **MID-SEMESTER ASSESSMENT CARD**. blueprint of a Report card or Assessment Card but based on all the data and information that we collected from our mid-semester marksheet for Advanced Programming and Database and Management systems. It is similar to a report card but we have tweaked it a little by adding in various parameters of comparison and reasoning such as printing the ranks of all the students whose viva was taken by a particular faculty member or the option to see which student has scored the most in the chosen subject.

MOTIVATION FOR CHOOSING THE TOPIC:

We wanted to incorporate database management systems and programming in such a way that it can be useful in a realistic scenario.
While we were choosing our topic, it was the time when we were being presented with our midsemester marks. With a class strength of 87, it is quite difficult to manage the huge amount of data that is presented to us in the form of marks. We thought of blending these two ideas together in such a way that it can be helpful to those trying to compare or check.
We thought of making a report card system that would help students find out potentially what their ranks in class are, how well they have performed in viva and under which faculty their marks were the highest. This project was made keeping in mind the real-time and real-world use of database systems and programming languages.

APPROACH FOR SOLVING THE PROBLEM:

This project is an amalgam of database systems and programming so methods of incorporating them together was our first plan of action.
Using a mind-map we first broke down the project to its various components, mentioning all of its parameters and attributes. We then found ways in which every aspect can be made user friendly from data collection to date presentation.
Most of our planning went into assuring that both the frontend and backend were in sync with one another.

MID-SEMESTER ASSESSMENT CARD

ADVANCED PROGRAMMING + DATABASE MANAGEMENT SYSTEMS

ANY DIFFICULTIES FACED (AND HOW WE SOLVED THEM):

Choosing between the two programming languages was quite a task at first. On one hand we had C++ which was familiar and easy to use and on the other hand we had Python which was advanced, came with a lot of elaborate in-built functions and is easier to connect with backend. In the end, sophistication won against familiarity and ended up choosing Python as our programming language.
During execution we realized that there were quite many glitches in connecting frontend and backend but with the help of a few YouTube videos and help from our faculty members we managed to resolve that.
We also found it slightly laborious to create every single one of the 19 cases one by one but on the suggestion of one of our team members we managed to combine a lot of them together and reduce redundancy and time of execution.
In the middle of our coding, we realized that if there were multiple top scorers for a subject, our code was only printing the one with the smallest roll number. The code and backend weren't supporting multiple answers and to solve this we had to learn many new database commands and implement it.

NEW THINGS WE LEARNT THROUGH THIS PROJECT:

We learnt how to work with backend and frontend simultaneously. We got to practice all the theory and knowledge we had acquired this semester in a more realistic and practical way. Python wasn't our strongest forte before this project but after having worked on this code, we are much more familiarized and comfortable with using it.

IMPLEMENTATION DETAILS:

For our project, we implemented Python and SQL using the software - Visual Studio Code and MySQL. We mainly used commands from TCL, DML and DCL under SQL.
In our code, we have started off by prompting the user to choose among the multiple options given to manipulate the students' marks information. On choosing one of the criterions, they are taken to the function that handles that particular option. There are many sub-parts to each option, thus giving the user a variety of alternatives to work with.
We have allowed the user to store, access, delete and manipulate students' mark's data, to retrieve the highest scorer in each subject, and the ranks according to a particular viva teacher.

DATABASE DESIGN

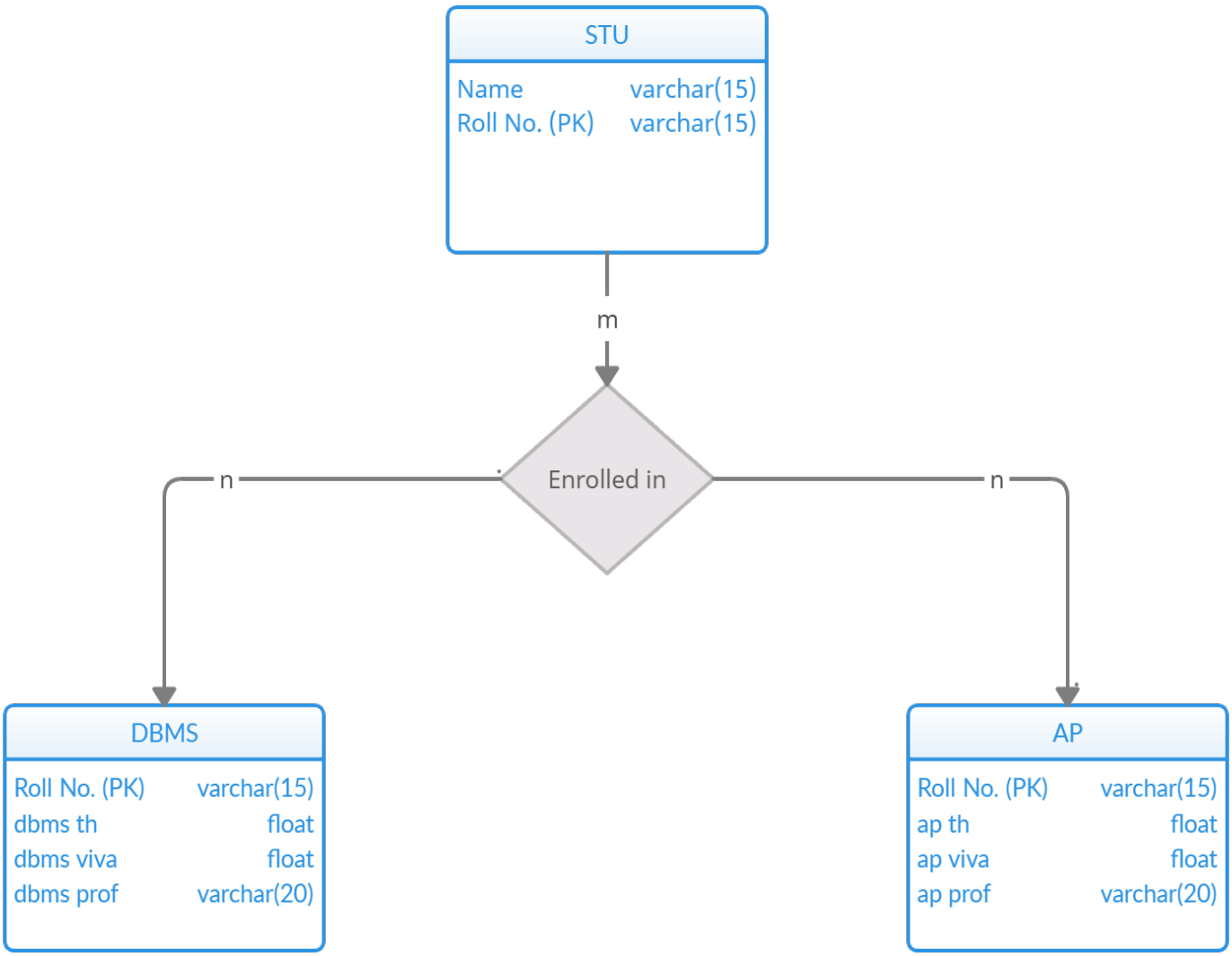


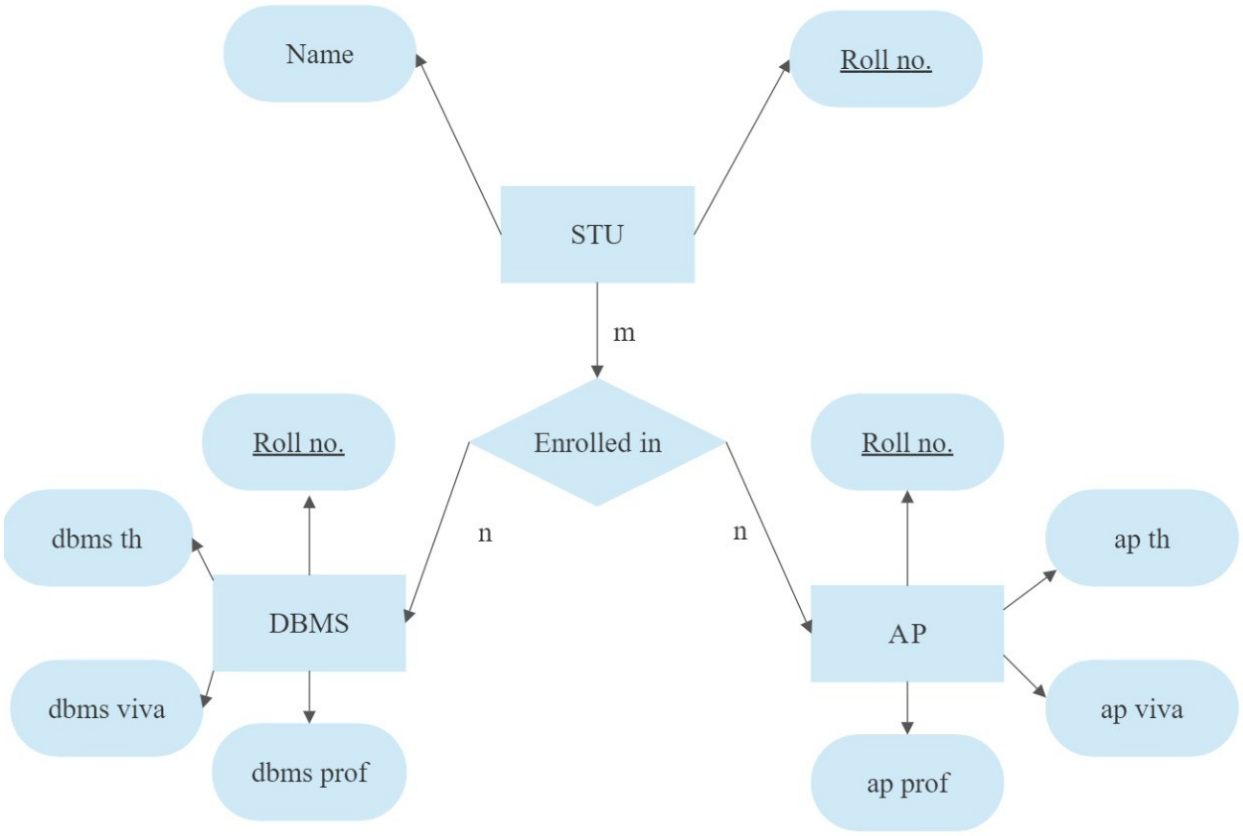
TABLE DESCRIPTION

STU Table	
Roll no.	varchar
Name	varchar

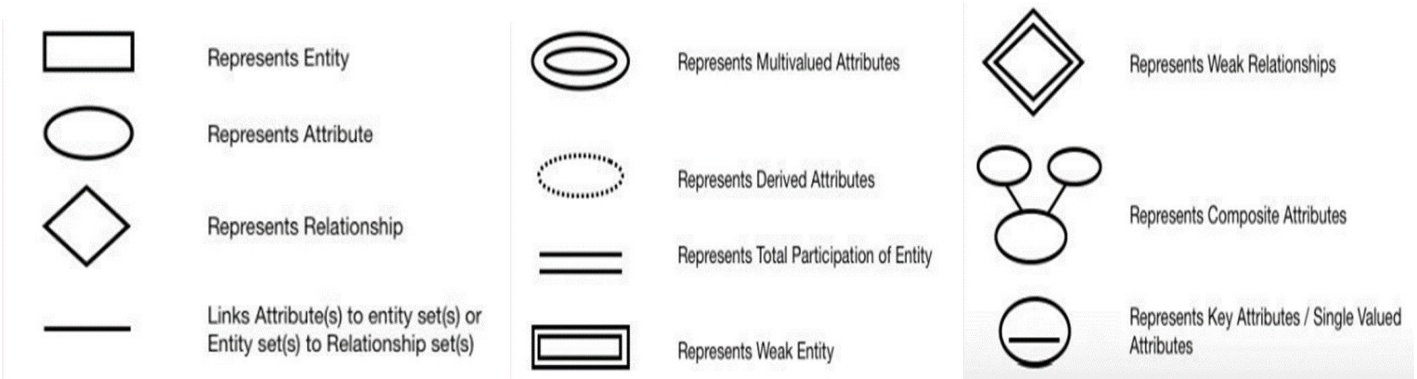
DBMS Table	
Roll no.	varchar
dbms th	float
dbms viva	float
dbms prof	varchar

AP Table	
Roll no.	varchar
ap th	float
ap viva	float
ap prof	varchar

ENTITY RELATIONSHIP



SYMBOL MEANING



CODE

```
import mysql.connector as driver
import sys
def main_menu():
    loop="y"
    while loop=="y":
        print("\n.....USER MENU.....")
        print("1. CREATE DATABASE")
        print("2. SHOW DATABASES")
        print("3. CREATE STUDENT TABLE")
        print("4. CREATE Advanced Programming Table")
        print("5. CREATE DataBase Management System Table")
        print("6. SHOW TABLES")
        print("7. ADD Student's Record")
        print("8. ADD STUDENT'S Advanced Programming Record")
        print("9. ADD STUDENT'S DataBase Management System Record")
        print("10. UPDATE  Advanced Programming Record")
        print("11. UPDATE DataBase Management System Record")
        print("12. DELETE Advanced Programming Record")
        print("13. DELETE DataBase Management System Record")
        print("14. SEARCH Advanced Programming Record")
        print("15. SEARCH DataBase Management System Record")
        print("16. DISPLAY Advanced Programming Record")
        print("17. DISPLAY DataBase Management System Record")
        print("18. DISPLAY Advanced Programming Topper Record")
        print("19. DISPLAY DataBase Management System Topper
Record")
        print("20. QUIT")
        print()
        choice=int(input("Enter the choice (1-20) : "))
        if(choice==1):
            create_database()
        elif(choice==2):
            show_databases()
```

```
elif(choice==3):
    create_stud_table()
elif(choice==4):
   create_ap_table()
elif(choice==5):
    create_dbms_table()
elif(choice==6):
     show_tables()
elif(choice==7):
    insert_stud_record()
elif(choice==8):
    insert_ap_record()
elif(choice==9):
    insert_dbms_record()
elif(choice==10):
    update_ap_record()
elif(choice==11):
    update_dbms_record()
elif(choice==12):
    delete_ap_record()
elif(choice==13):
    delete_dbms_record()
elif(choice==14):
    search_ap_record()
elif(choice==15):
    search_dbms_record()
elif(choice==16):
    display_ap_record()
elif(choice==17):
    display_dbms_record()
elif(choice==18):
    display_ap_top_record()
elif(choice==19):
    display_dbms_top_record()

elif(choice==20):
```

```
            break
        else:
            print("Wrong Choice.")
        loop=input("\nDo you want to continue?(y or n)")
    else:
        sys.exit()

def create_database():

con=driver.connect(host="localhost",user="root",passwd="622001",port=
"3306",charset='utf8')
    if con.is_connected():
        print("Successfully Connected")
    cur=con.cursor()
    cur.execute('create database if not exists test')
    print()
    print("Database Created")
    con.close()

def show_databases():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',port="3306")
    if con.is_connected():
        print("Successfully Connected")
    cur=con.cursor()
    cur.execute('show databases')
    for i in cur:
        print(i)
    con.close()

def create_stud_table():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    if con.is_connected():
```

```
        print("Successfully Connected")
    cur=con.cursor()
    cur.execute('create table if not exists stu(rollno varchar(15) primary
key, name varchar(15))')
    print()
    print("Table Created -> stu")
    cur.execute('DESC stu')
    print("+-------------|-------------|----------+")
    print("+Column Name  |DataType(Size)|NULL       |")
    print("+-------------|-------------|----------+")

    for i in cur:
        print('|{0:12} | {1:12} | {2:10}|'.format(i[0],i[1].decode('UTF-
8'),i[2]))
    print("+-------------|-------------|----------+")
    con.close()

def create_ap_table():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    if con.is_connected():
        print("Successfully Connected")
    cur=con.cursor()
    cur.execute('create table if not exists AP(rollno varchar(15) primary
key, apth float, apviva float, approf varchar(20))')
    print()
    print("Table Created -> AP")
    cur.execute('DESC AP')
    print("+-------------|-------------|----------+")
    print("+Column Name  |DataType(Size)|NULL       |")
    print("+-------------|-------------|----------+")

    for i in cur:
        print('|{0:12} | {1:12} | {2:10}|'.format(i[0],i[1].decode('UTF-
8'),i[2]))
```

```
    print("+------------|-------------|----------+")
    con.close()

def create_dbms_table():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    if con.is_connected():
        print("Successfully Connected")
    cur=con.cursor()
    cur.execute('create table if not exists DBMS(rollno varchar(15) primary
key, dbmsth float, dbmsviva float, dbmsprof varchar(20))')
    print()
    print("Table Created -> DBMS")
    cur.execute('DESC DBMS')
    print("+------------|-------------|----------+")
    print("+Column Name  |DataType(Size)|NULL       |")
    print("+------------|-------------|----------+")

    for i in cur:
        print('|{0:12} | {1:12} | {2:10}|'.format(i[0],i[1].decode('UTF-
8'),i[2]))
    print("+------------|-------------|----------+")
    con.close()

def show_tables():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    if con.is_connected():
        print("Successfully Connected")
    cur=con.cursor()
    cur.execute('show tables')
    for i in cur:
        print(i)
    con.close()
```

```
def insert_stud_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    if con.is_connected():
        print("Successfully Connected")
        cur=con.cursor()
        rollno=(input("ENTER student rollno : "))
        NAME=input("ENTER Name OF student : ")

        query1="INSERT INTO stu(rollno,name)
VALUES('{}','{}')".format(rollno,NAME)
        cur.execute(query1)
        con.commit()
        print('Record Inserted')
        con.close()
    else:
        print("Error : Not Connected")

def insert_ap_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    if con.is_connected():
        print("Successfully Connected")
        cur=con.cursor()
        rollno=(input("ENTER student rollno : "))
        apth=eval(input("enter marks in theory : "))
        apviva=eval(input("enter viva marks : "))
        approf=input("enter name of the professor who took viva in
Advanced Programming : ")

        query1="INSERT INTO AP(rollno,apth,apviva,approf)
VALUES('{}',{},{},'{}')".format(rollno,apth,apviva,approf)
        cur.execute(query1)
```

```
        con.commit()
        print('Record Inserted')
        con.close()
    else:
        print("Error : Not Connected")

def insert_dbms_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    if con.is_connected():
        print("Successfully Connected")
        cur=con.cursor()
        rollno=(input("ENTER student rollno : "))
        dbmsth=eval(input("enter marks in theory : "))
        dbmsviva=eval(input("enter viva marks : "))
        dbmsprof=input("enter name of the professor who took viva in
DataBase Management System : ")

        query1="INSERT INTO DBMS(rollno,dbmsth,dbmsviva,dbmsprof)
VALUES('{}',{},{},'{}')".format(rollno,dbmsth,dbmsviva,dbmsprof)
        cur.execute(query1)
        con.commit()
        print('Record Inserted')
        con.close()
    else:
        print("Error : Not Connected")

def update_ap_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    cur=con.cursor()

    rollno=(input("enter Student rollno : "))
    apth=eval(input("enter marks in theory to be updated : "))
```

```
    apviva=eval(input("enter viva marks to be updated : "))


    query1="update AP set rollno=%s apth=%s , apviva=%s where rollno=
%s" %(rollno,apth,apviva)
    cur.execute(query1)
    con.commit()
    print("Record Updated")
    con.close()

def update_dbms_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    cur=con.cursor()

    rollno=(input("enter Student rollno : "))
    dbmsth=eval(input("enter marks in theory to be updated : "))
    dbmsviva=eval(input("enter viva marks to be updated : "))


    query1="update DBMS set rollno=%s dbmsth=%s , dbmsviva=%s
where rollno=%s" %(rollno,dbmsth,dbmsviva)
    cur.execute(query1)
    con.commit()
    print("Record Updated")
    con.close()

def delete_ap_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    cur=con.cursor()
    d=(input("Enter student rollno for deleting record : "))
    query1="delete from AP where rollno={0}".format(d)
    cur.execute(query1)
```

```
    con.commit()
    print("Record Deleted")
    con.close()

def delete_dbms_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    cur=con.cursor()
    d=(input("Enter student rollno for deleting record : "))
    query1="delete from DBMS where rollno={0}".format(d)
    cur.execute(query1)
    con.commit()
    print("Record Deleted")
    con.close()

def search_ap_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    cur=con.cursor()
    print("enter the choice according to which you have to search in AP
record: ")
    print("1. According to roll number")
    print("2. According TO Viva Professor's Name")
    print("3. According to theory marks")
    print("4. According to viva marks")
    print()
    choice=int(input("ENTER THE CHOICE (1-4) : "))
    if choice==1:
        d=(input("Enter student rollno which you want to search : "))
        query1="select * from AP where rollno=%s" %(d)
    elif choice==2:
        name=input("Enter Viva Professor's Name which you want to
search : ")
        query1="select * from AP where name='%s'" %(name)
```

```
    elif choice==3:
        th_mark=float(input("Enter theory marks which you want to
search : "))
        query1="select * from AP where marks=%s" %(th_mark)
    elif choice==4:
        viva_mark=float(input("Enter viva marks which you want to
search : "))
        query1="select * from AP where marks=%s" %(viva_mark)
    else:
        print("Wrong Choice")
    cur.execute(query1)
    rec=cur.fetchall()
    count=cur.rowcount
    print("Total no. of records found : ",count)
    for i in rec:
        print(i)
    print("Record Searched")
    con.close()

def search_dbms_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    cur=con.cursor()
    print("enter the choice according to which you have to search in DBMS
record: ")
    print("1. According to roll number")
    print("2. According TO Viva Professor's Name")
    print("3. According to theory marks")
    print("4. According to viva marks")
    print()
    choice=int(input("ENTER THE CHOICE (1-4) : "))
    if choice==1:
        d=(input("Enter student rollno which you want to search : "))
        query1="select * from DBMS where rollno=%s" %(d)
    elif choice==2:
```

```
        name=input("Enter Viva Professor's Name which you want to
search : ")
        query1="select * from DBMS where name='%s'" %(name)
    elif choice==3:
        th_mark=float(input("Enter theory marks which you want to
search : "))
        query1="select * from DBMS where marks=%s" %(th_mark)
    elif choice==4:
        viva_mark=float(input("Enter viva marks which you want to
search : "))
        query1="select * from DBMS where marks=%s" %(viva_mark)
    else:
        print("Wrong Choice")
    cur.execute(query1)
    rec=cur.fetchall()
    count=cur.rowcount
    print("Total no. of records found : ",count)
    for i in rec:
        print(i)
    print("Record Searched")
    con.close()

def display_ap_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    if con.is_connected():
        print("Successfully Connected")
        cur=con.cursor()
        cur.execute('select * from AP')
        rec=cur.fetchall()
        count=cur.rowcount
        print("Advanced Programming Record is as follows: ")
        print()
        print("+--------------|-----------------|---------------|------------------
+")
```

```
      print("|  Roll Number  |   Theory Marks   |   Viva Marks   |   Viva
Professor   |")
      print("+--------------|----------------|--------------|------------------+")
       for i in rec:
          print('|{0:^11} | {1:^16} | {2:^15}|
{3:^20}|'.format(i[0],i[1],i[2],i[3]))
      print("+--------------|----------------|--------------|------------------+")
       print("|                    Total no. of records are : ",count,"                  |")
     print("+-------------------------------------------------------------------+")
       con.close()
   else:
      print("Error : Database Connection is not success")

def display_dbms_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
   if con.is_connected():
      print("Successfully Connected")
      cur=con.cursor()
      cur.execute('select * from DBMS')
      rec=cur.fetchall()
      count=cur.rowcount
      print("DataBase Management System Record Is As Follows: ")
      print()
      print("+-------------|----------------|--------------|------------------+")
      print("|  Roll Number |   Theory Marks   |   Viva Marks   |   Viva
Professor   |")
      print("+-------------|----------------|--------------|------------------+")
      for i in rec:
          print('|{0:^13} | {1:^16} | {2:^15}|
{3:^20}|'.format(i[0],i[1],i[2],i[3]))
      print("+-------------|----------------|--------------|------------------+")
      print("|                    Total no. of records are : ",count,"                  |")
      print("+-------------------------------------------------------------------+")
```

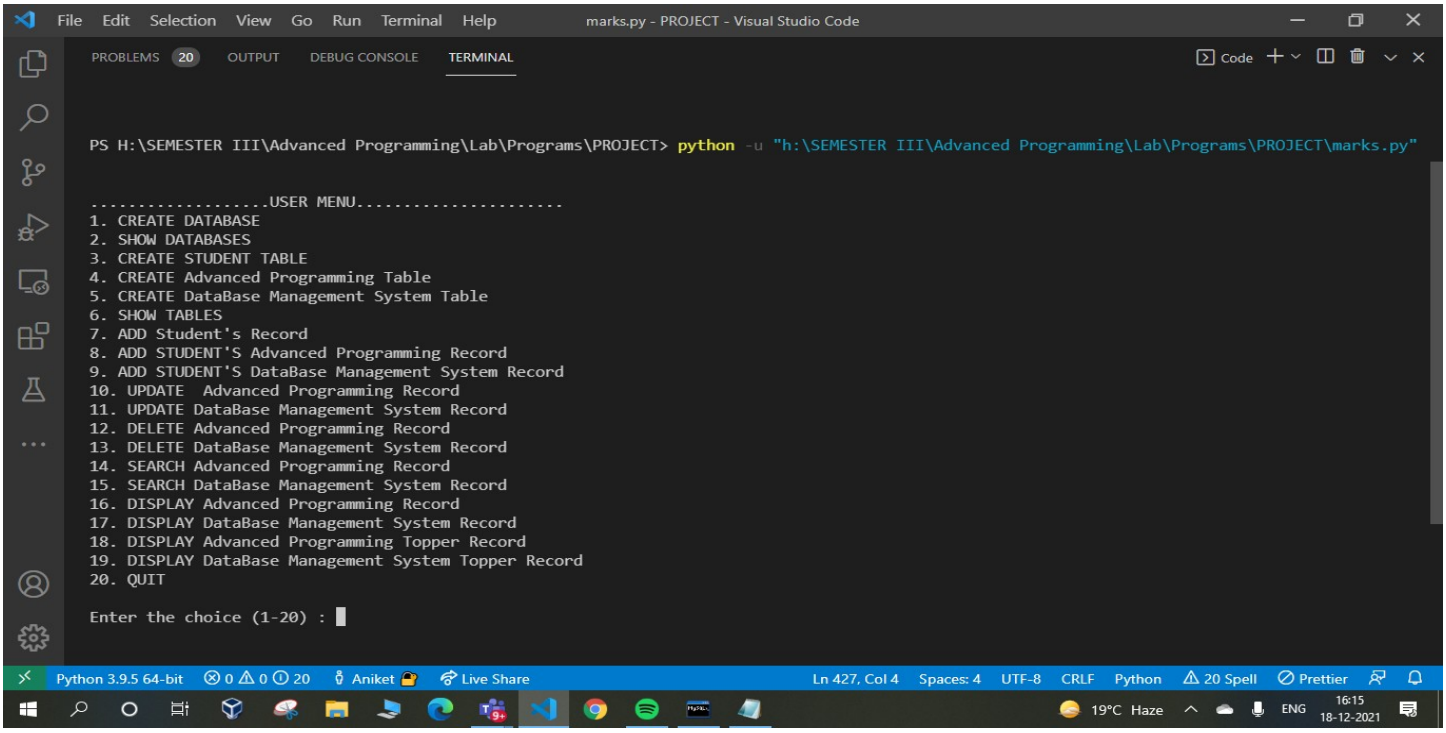
```
 con.close()
    else:
        print("Error : Database Connection is not success")


def display_ap_top_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    if con.is_connected():
        print("Successfully Connected")
        cur=con.cursor()
        cur.execute('select name from tes1, rollno from AP where
SUM(apth,apviva) is MAX')

def display_ap_top_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    cur=con.cursor()
    print("Enter the choice according to which you have to search topper in
Advanced Programming: ")

    print("1. According to Viva Professor's Name")
    print("2. According to theory marks")
    print("3. According to viva marks")
    print()
    choice=int(input("ENTER THE CHOICE (1-3) : "))

    if choice==1:
        name=input("Enter Viva Professor's Name whose highest awarded
marks you want to search : ")
        query1="select  stu.name, ap.rollno, ap.apviva from AP, stu where
stu.rollno = ap.rollno and ap.approf ='%s'" %(name)
    elif choice==2:
```

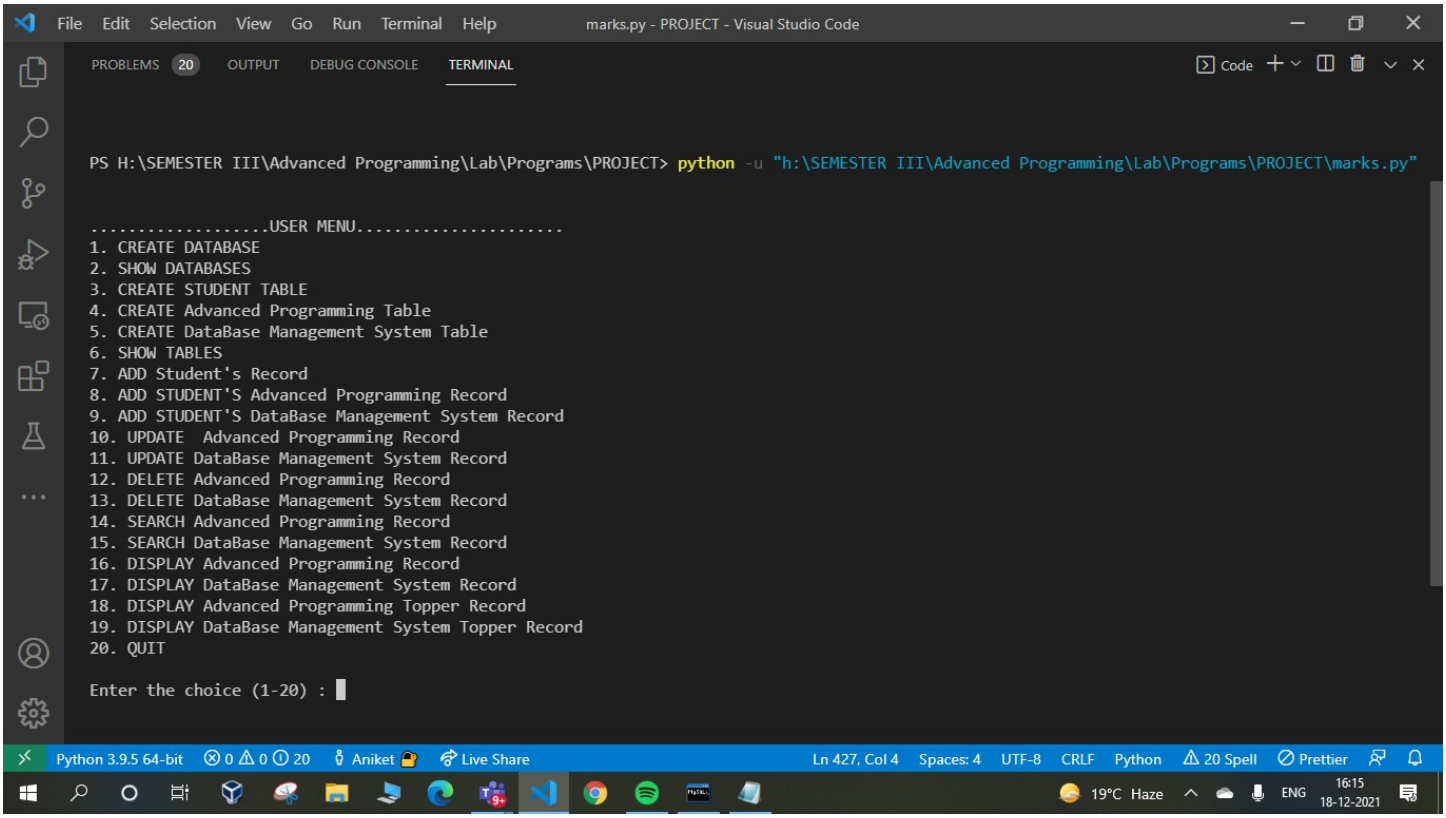
```
        query1="SELECT  stu.name, ap.rollno,ap.apth FROM AP INNER
JOIN stu ON stu.rollno=ap.rollno where apth=(select max(apth) from
AP)"
    elif choice==3:

        query1="SELECT  stu.name, ap.rollno,ap.apviva FROM AP
INNER JOIN stu ON stu.rollno=ap.rollno where apviva=(select
max(apviva) from AP) "
    else:
        print("Wrong Choice")
    cur.execute(query1)
    rec=cur.fetchall()
    count=cur.rowcount
    print("Advanced Programming Record Is As Follows: ")
    print()
    print("+---------------|---------------|---------------+")
    print("|      Name      |  Roll Number   |   Max Marks    |")
    print("+---------------|---------------|---------------+")
    for i in rec:
        print('|{0:^15} |{1:^15}   {2:^14} | '.format(i[0],i[1],i[2]))
    print("+---------------|---------------|---------------+")
    print("|            Total records are : ",count,"             |")
    print("+-----------------------------------------------+")
    con.close()

def display_dbms_top_record():

con=driver.connect(host='localhost',user='root',passwd='622001',charset='
utf8',database='test',port="3306")
    cur=con.cursor()
    print("Enter the choice according to which you have to search topper in
DataBase Management System: ")

    print("1. Rank Card According to Viva Professor's Name")
    print("2. According to theory marks")
    print("3. According to viva marks")
```

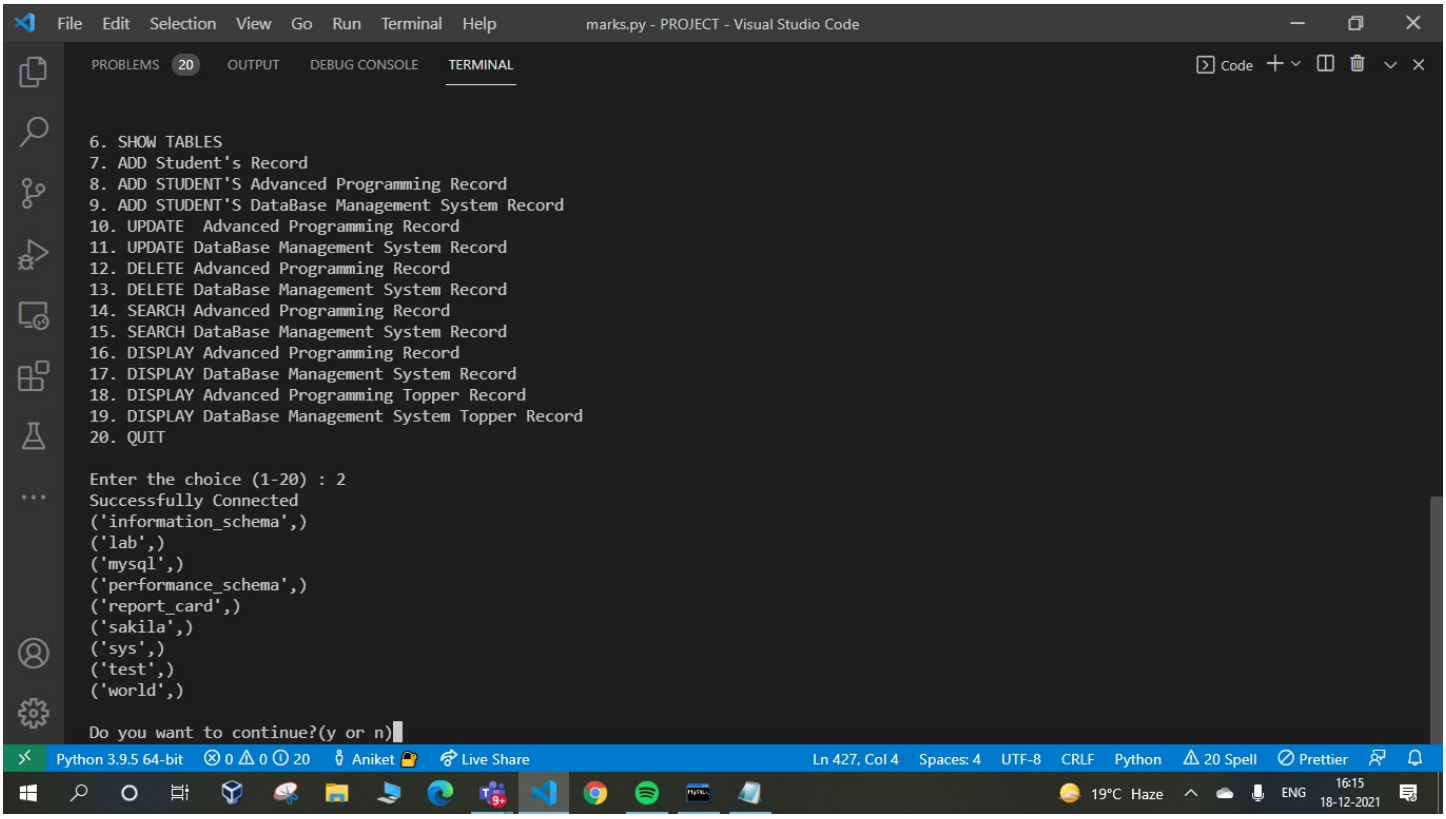
```
    print()
    choice=int(input("ENTER THE CHOICE (1-3) : "))

    if choice==1:
        name=input("Enter Viva Professor's Name whose highest awarded
marks you want to search : ")
        query1="select  stu.name, dbms.rollno, dbms.dbmsviva from dbms,
stu where stu.rollno = dbms.rollno and dbms.dbmsprof = '%s'" %(name)
    elif choice==2:

        query1=" SELECT  stu.name, dbms.rollno, dbms.dbmsth FROM
dbms INNER JOIN stu ON stu.rollno=dbms.rollno where dbmsth=(select
max(dbmsth) from dbms)"
    elif choice==3:

        query1="SELECT  stu.name, dbms.rollno, dbms.dbmsviva FROM
dbms INNER JOIN stu ON stu.rollno=dbms.rollno where
dbmsviva=(select max(dbmsviva) from dbms)"
    else:
        print("Wrong Choice")
    cur.execute(query1)
    rec=cur.fetchall()
    count=cur.rowcount
    print("DataBase Management System Record Is As Follows: ")
    print()
    print("+---------------|---------------|---------------+")
    print("|  Name         |  Roll Number   |   Max Marks    |")
    print("+---------------|---------------|---------------|")
    for i in rec:
         print('|{0:^15} |{1:^15} | {2:^14} | '.format(i[0],i[1],i[2]))
    print("+---------------|---------------|---------------+")
    print("|             Total records are : ",count,"             |")
    print("+-----------------------------------------------+")
    con.close()
```

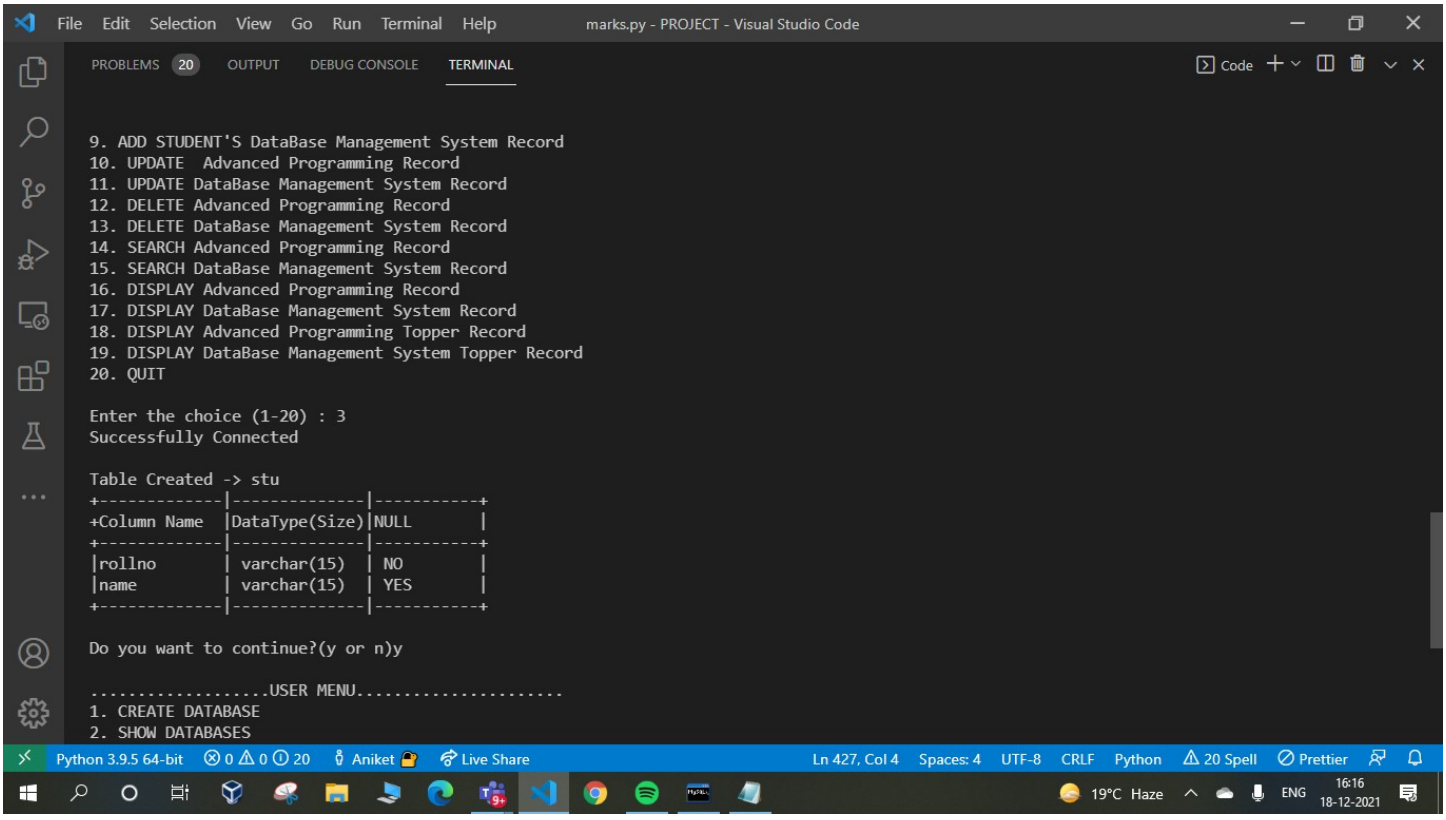
```
if __name__ == "__main__":
    main_menu()
```

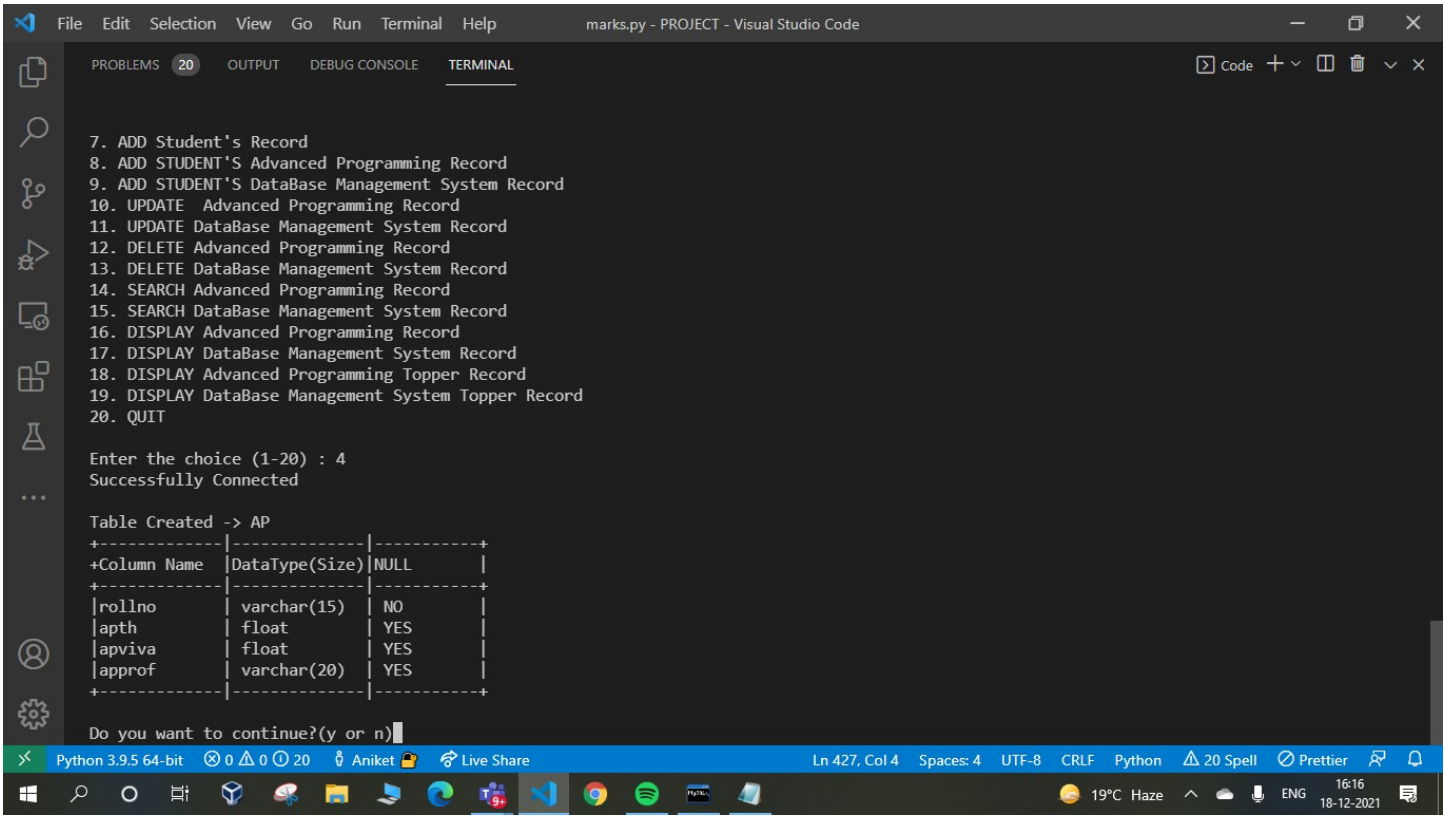
OUTPUT SCREENSHOTS

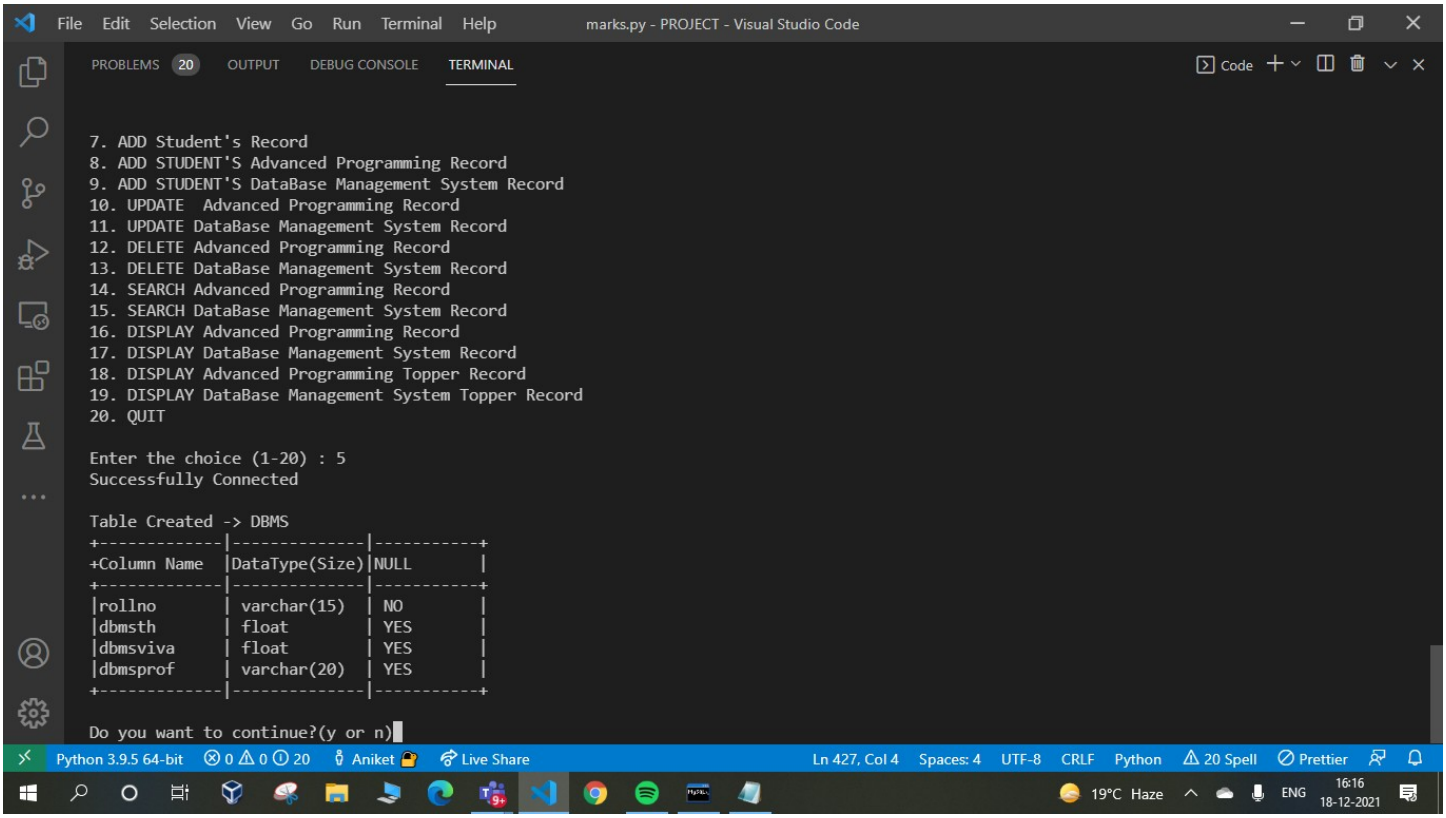


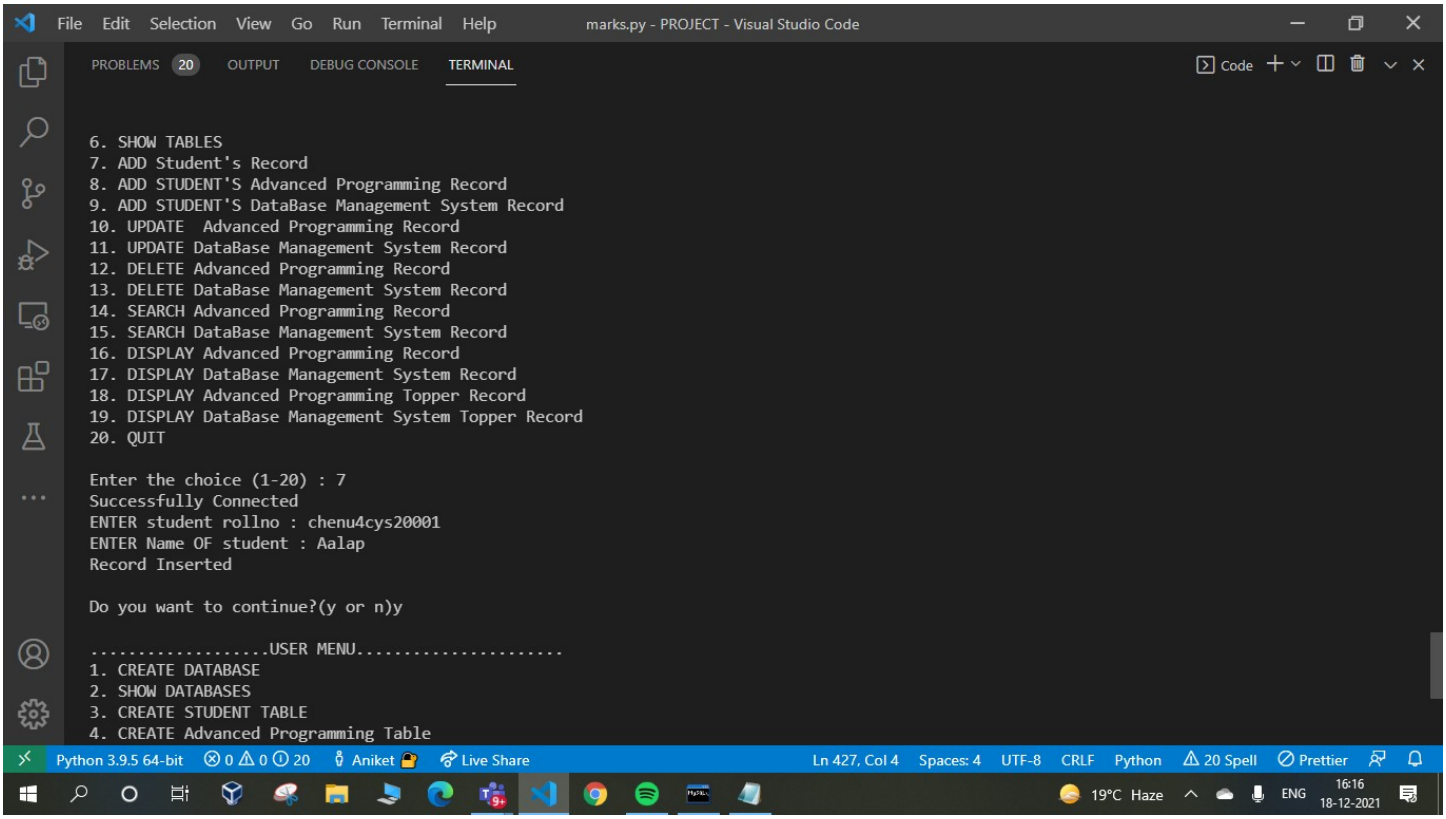


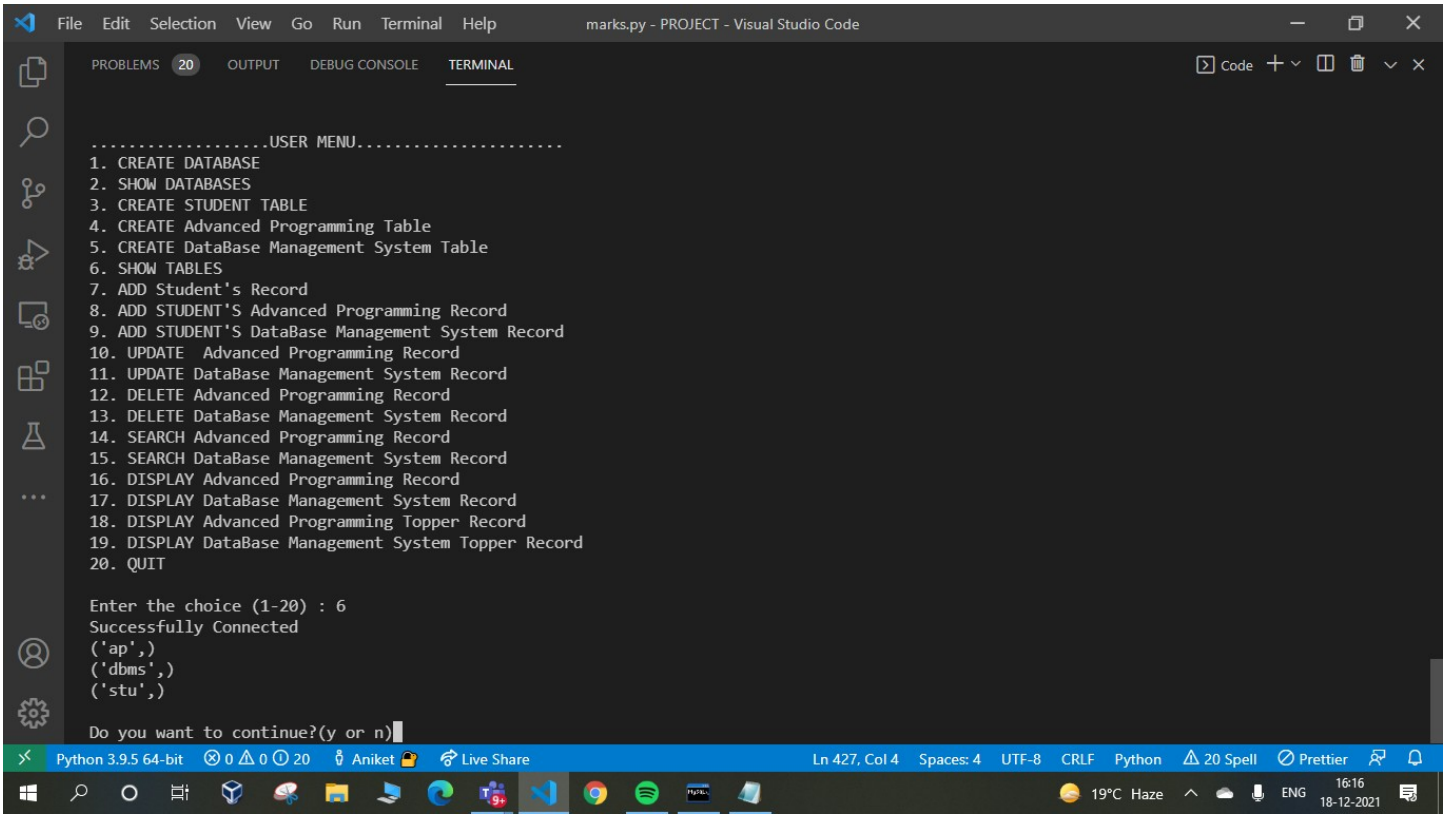

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
6. SHOW TABLES
7. ADD Student's Record
8. ADD STUDENT'S Advanced Programming Record
9. ADD STUDENT'S DataBase Management System Record
10. UPDATE Advanced Programming Record
11. UPDATE DataBase Management System Record
12. DELETE Advanced Programming Record
13. DELETE DataBase Management System Record
14. SEARCH Advanced Programming Record
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 2
Successfully Connected
('information_schema',)
('lab',)
('mysql',)
('performance_schema',)
('report_card',)
('sakila',)
('sys',)
('test',)
('world',)
Do you want to continue?(y or n)
Python 3.9.5 64-bit
Aniket
Live Share
Ln 427, Col 4
Spaces: 4
UTF-8
CRLF
Python
20 Spell
Prettier
19°C Haze
ENG
16:15
18-12-2021

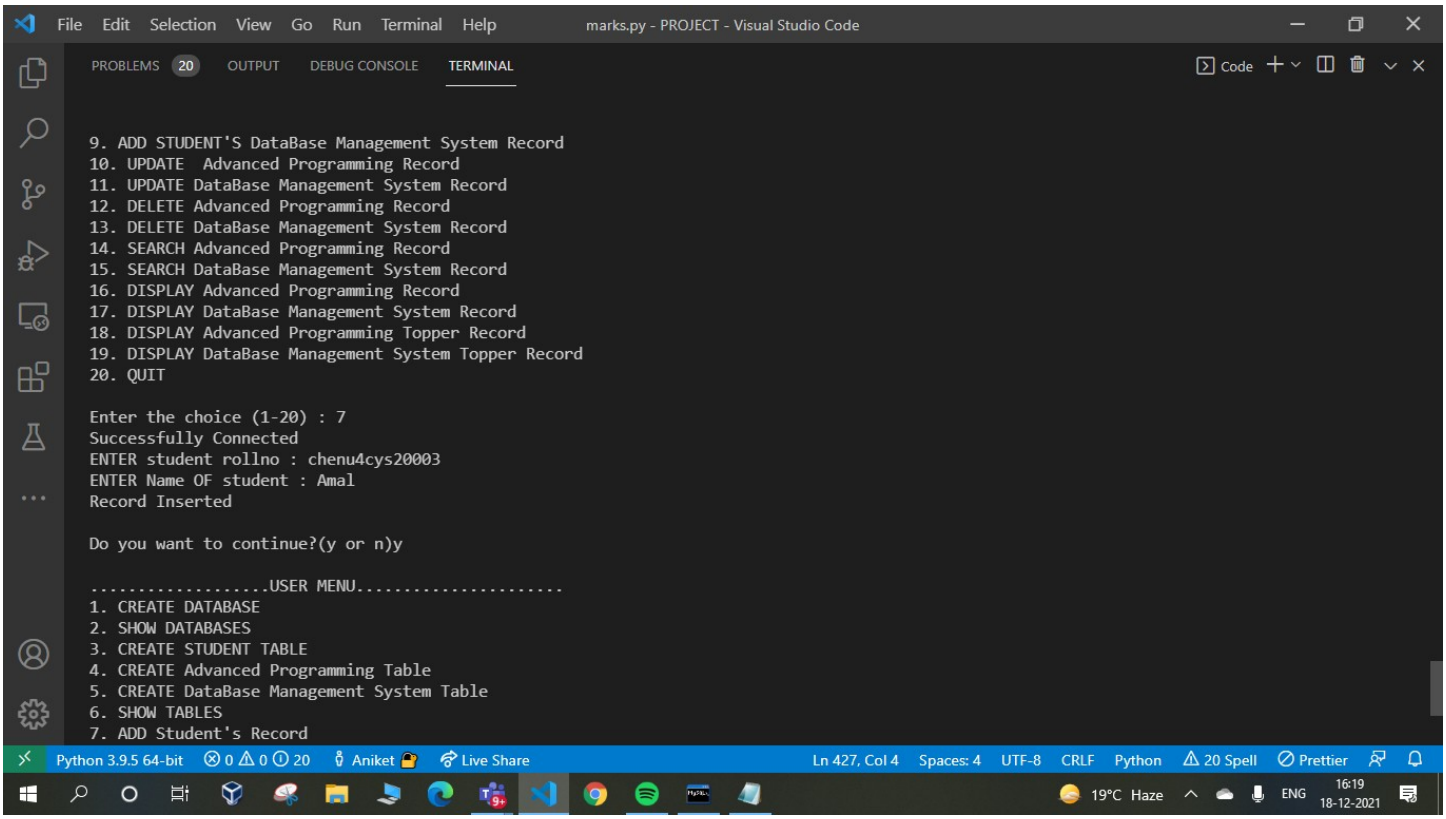

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
9. ADD STUDENT'S DataBase Management System Record
10. UPDATE Advanced Programming Record
11. UPDATE DataBase Management System Record
12. DELETE Advanced Programming Record
13. DELETE DataBase Management System Record
14. SEARCH Advanced Programming Record
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 3
Successfully Connected
Table Created -> stu
+-------------|--------------|-----------+
+Column Name |DataType(Size)|NULL |
+-------------|--------------|-----------+
|rollno | varchar(15) | NO |
|name | varchar(15) | YES |
+-------------|--------------|-----------+
Do you want to continue?(y or n)y
.....USER MENU.....
1. CREATE DATABASE
2. SHOW DATABASES
Python 3.9.5 64-bit
Aniket
Live Share
Ln 427, Col 4
Spaces: 4
UTF-8
CRLF
Python
20 Spell
Prettier
19°C Haze
ENG
16:16
18-12-2021

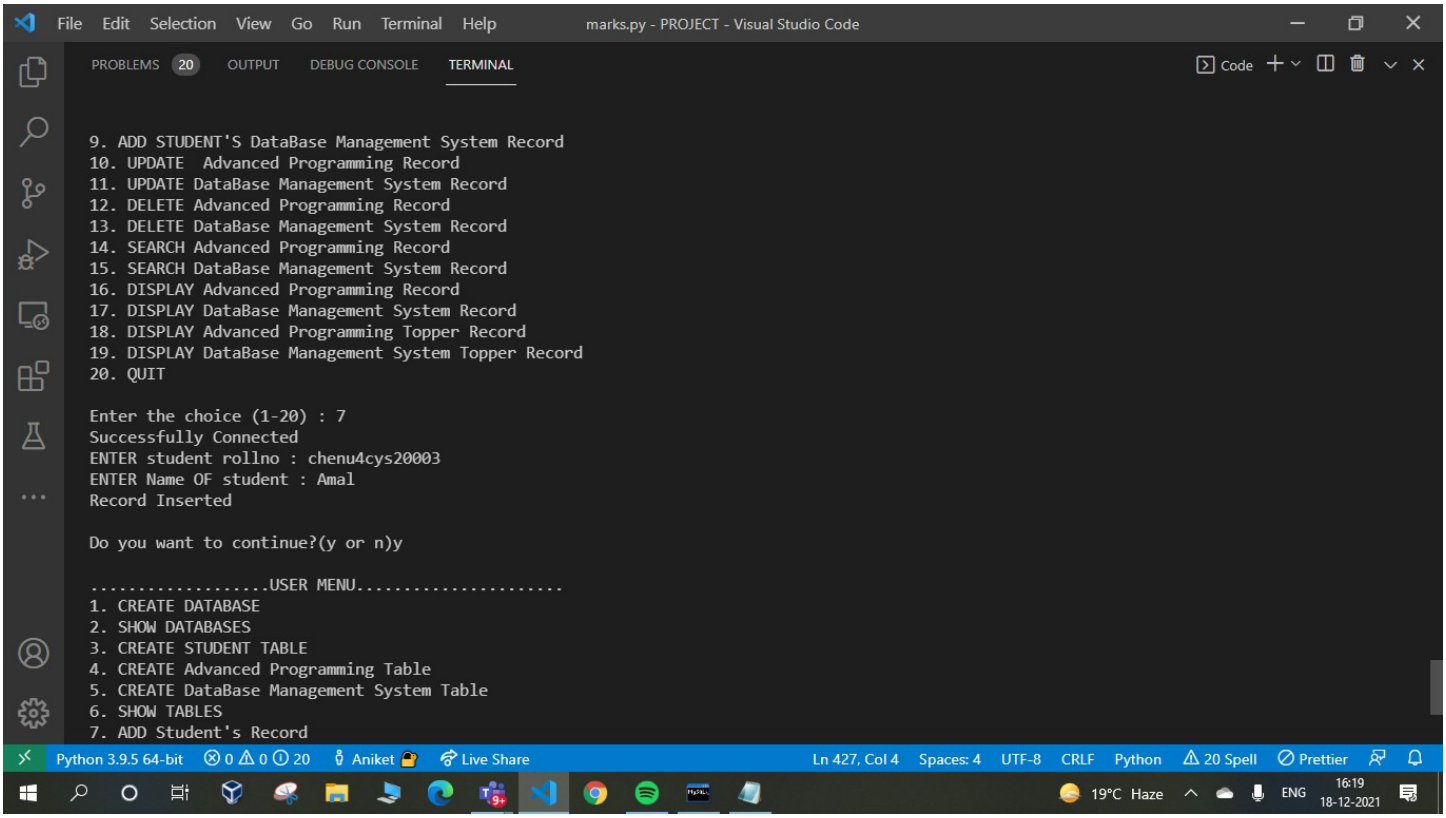

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
7. ADD Student's Record
8. ADD STUDENT'S Advanced Programming Record
9. ADD STUDENT'S DataBase Management System Record
10. UPDATE Advanced Programming Record
11. UPDATE DataBase Management System Record
12. DELETE Advanced Programming Record
13. DELETE DataBase Management System Record
14. SEARCH Advanced Programming Record
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 4
Successfully Connected
Table Created -> AP
+-------------|--------------|-----------+
+Column Name |DataType(Size)|NULL |
+-------------|--------------|-----------+
rollno	varchar(15)	NO
apth	float	YES
apviva	float	YES
approf	varchar(20)	YES
+-------------|--------------|-----------+
Do you want to continue?(y or n)
Python 3.9.5 64-bit
Aniket
Live Share
Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze
ENG
16:16
18-12-2021


File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
7. ADD Student's Record
8. ADD STUDENT'S Advanced Programming Record
9. ADD STUDENT'S DataBase Management System Record
10. UPDATE Advanced Programming Record
11. UPDATE DataBase Management System Record
12. DELETE Advanced Programming Record
13. DELETE DataBase Management System Record
14. SEARCH Advanced Programming Record
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 5
Successfully Connected
Table Created -> DBMS
+-------------|--------------|-----------+
+Column Name |DataType(Size)|NULL |
+-------------|--------------|-----------+
rollno	varchar(15)	NO
dbmsth	float	YES
dbmsviva	float	YES
dbmsprof	varchar(20)	YES
+-------------|--------------|-----------+
Do you want to continue?(y or n)
Python 3.9.5 64-bit
Aniket
Live Share
Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze
ENG
16:16
18-12-2021

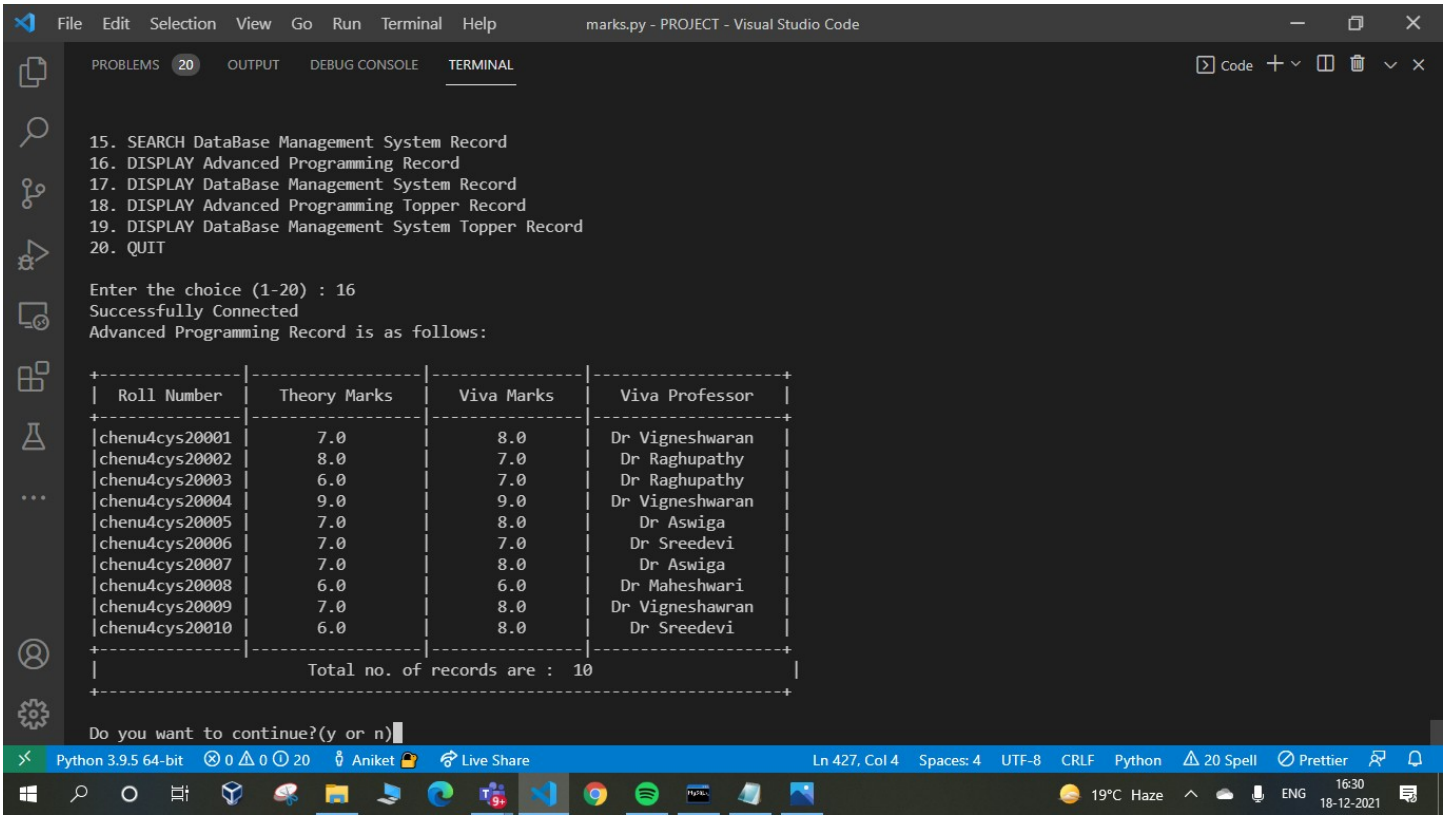

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
6. SHOW TABLES
7. ADD Student's Record
8. ADD STUDENT'S Advanced Programming Record
9. ADD STUDENT'S DataBase Management System Record
10. UPDATE Advanced Programming Record
11. UPDATE DataBase Management System Record
12. DELETE Advanced Programming Record
13. DELETE DataBase Management System Record
14. SEARCH Advanced Programming Record
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 7
Successfully Connected
ENTER student rollno : chenu4cys20001
ENTER Name OF student : Aalap
Record Inserted
Do you want to continue?(y or n)y
.....USER MENU.....
1. CREATE DATABASE
2. SHOW DATABASES
3. CREATE STUDENT TABLE
4. CREATE Advanced Programming Table
Python 3.9.5 64-bit ⊗0 ⚠0 ⓘ20 Aniket Live Share
Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze ENG 16:16 18-12-2021

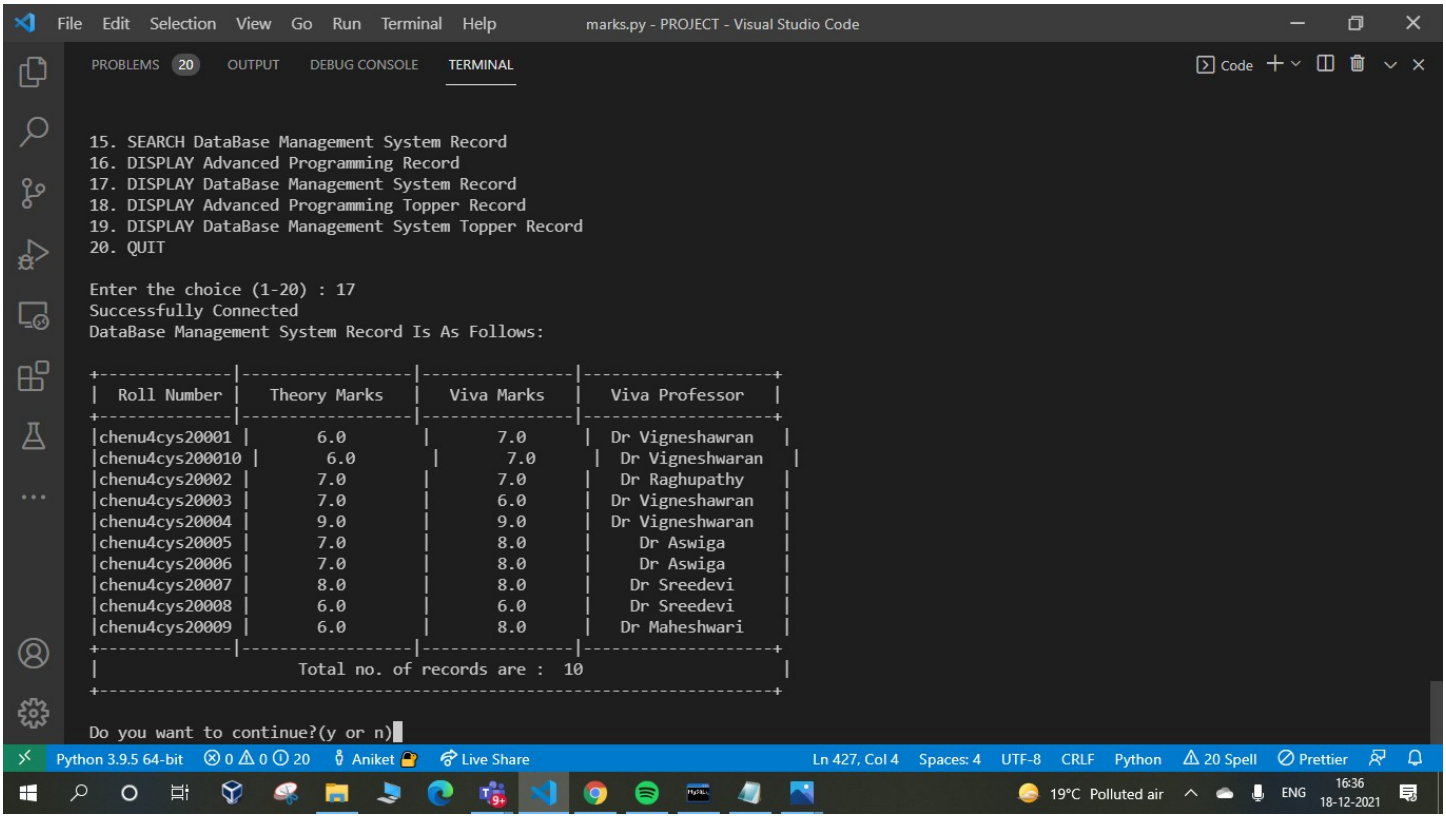

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
.....USER MENU.....
1. CREATE DATABASE
2. SHOW DATABASES
3. CREATE STUDENT TABLE
4. CREATE Advanced Programming Table
5. CREATE DataBase Management System Table
6. SHOW TABLES
7. ADD Student's Record
8. ADD STUDENT'S Advanced Programming Record
9. ADD STUDENT'S DataBase Management System Record
10. UPDATE Advanced Programming Record
11. UPDATE DataBase Management System Record
12. DELETE Advanced Programming Record
13. DELETE DataBase Management System Record
14. SEARCH Advanced Programming Record
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 6
Successfully Connected
('ap',)
('dbms',)
('stu',)
Do you want to continue?(y or n)
Python 3.9.5 64-bit ⊗0 ⚠0 ⓘ20 Aniket Live Share
Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze ENG 16:16 18-12-2021

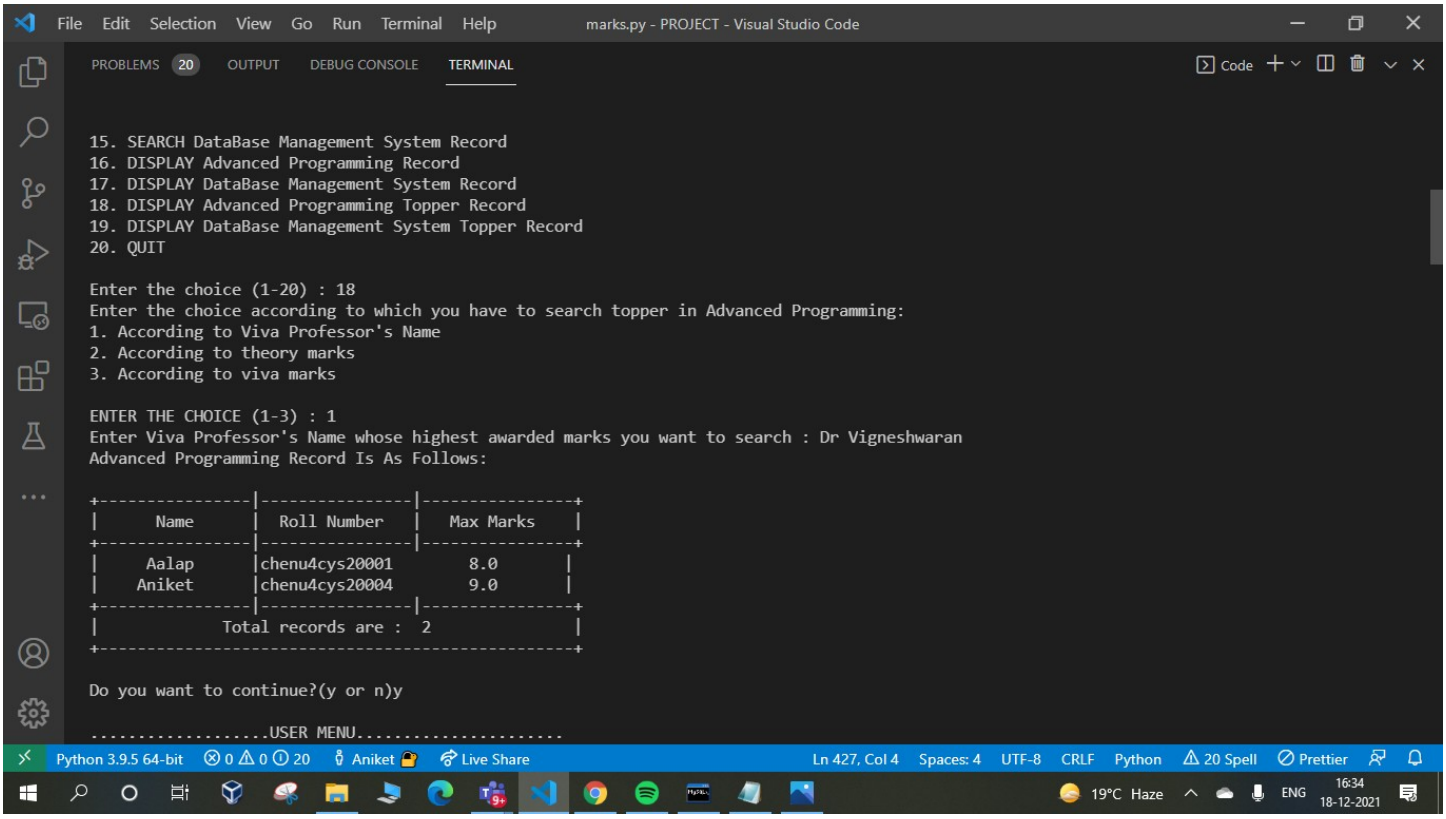

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
9. ADD STUDENT'S DataBase Management System Record
10. UPDATE Advanced Programming Record
11. UPDATE DataBase Management System Record
12. DELETE Advanced Programming Record
13. DELETE DataBase Management System Record
14. SEARCH Advanced Programming Record
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 7
Successfully Connected
ENTER student rollno : chenu4cys20003
ENTER Name OF student : Amal
Record Inserted
Do you want to continue?(y or n)y
.....USER MENU.....
1. CREATE DATABASE
2. SHOW DATABASES
3. CREATE STUDENT TABLE
4. CREATE Advanced Programming Table
5. CREATE DataBase Management System Table
6. SHOW TABLES
7. ADD Student's Record
Python 3.9.5 64-bit 0 0 20 Aniket Live Share
Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze ENG 16:19 18-12-2021

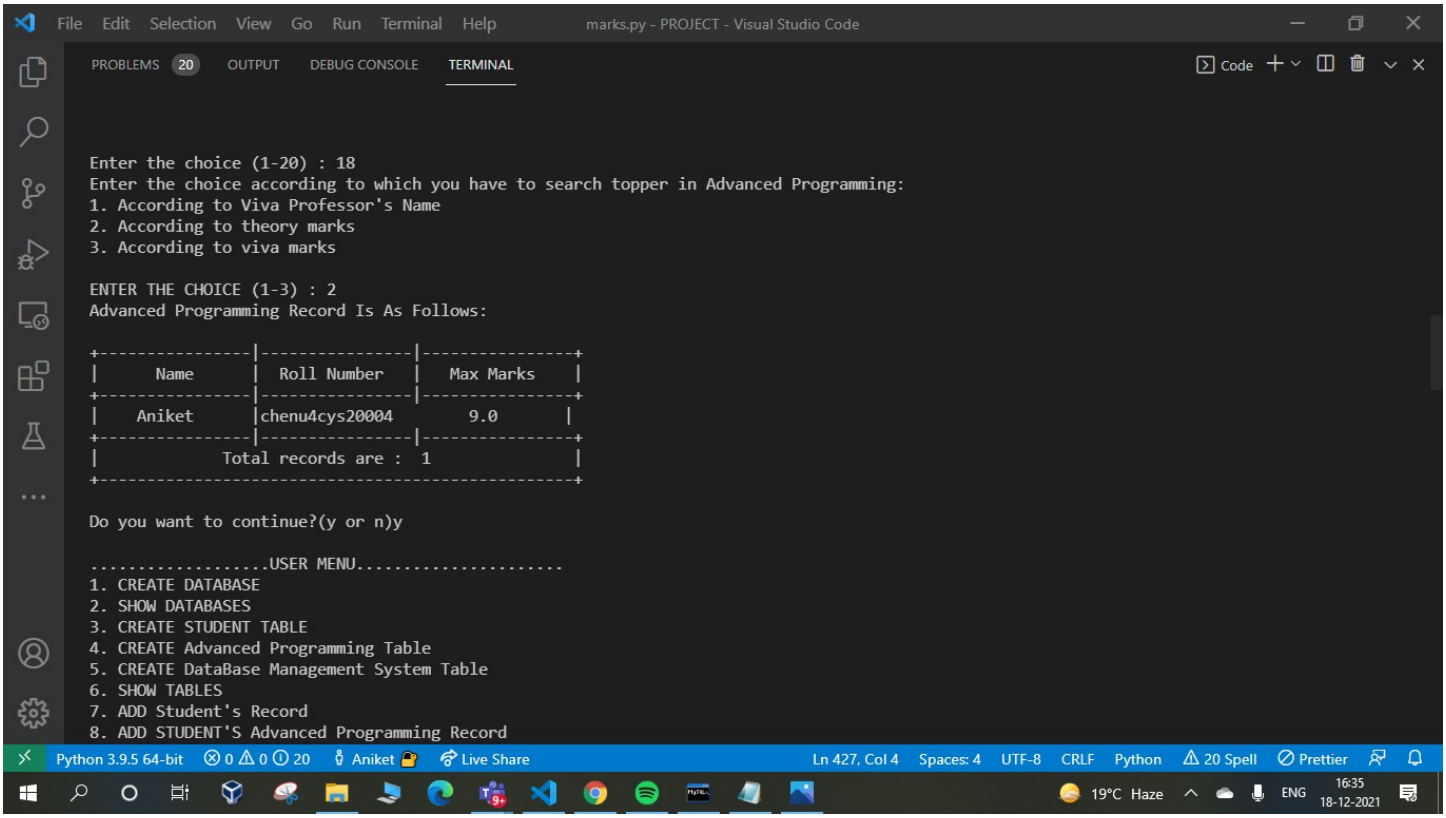

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
9. ADD STUDENT'S DataBase Management System Record
10. UPDATE Advanced Programming Record
11. UPDATE DataBase Management System Record
12. DELETE Advanced Programming Record
13. DELETE DataBase Management System Record
14. SEARCH Advanced Programming Record
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 7
Successfully Connected
ENTER student rollno : chenu4cys20003
ENTER Name OF student : Amal
Record Inserted
Do you want to continue?(y or n)y
.....USER MENU.....
1. CREATE DATABASE
2. SHOW DATABASES
3. CREATE STUDENT TABLE
4. CREATE Advanced Programming Table
5. CREATE DataBase Management System Table
6. SHOW TABLES
7. ADD Student's Record
Python 3.9.5 64-bit 0 0 20 Aniket Live Share
Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze ENG 16:19 18-12-2021

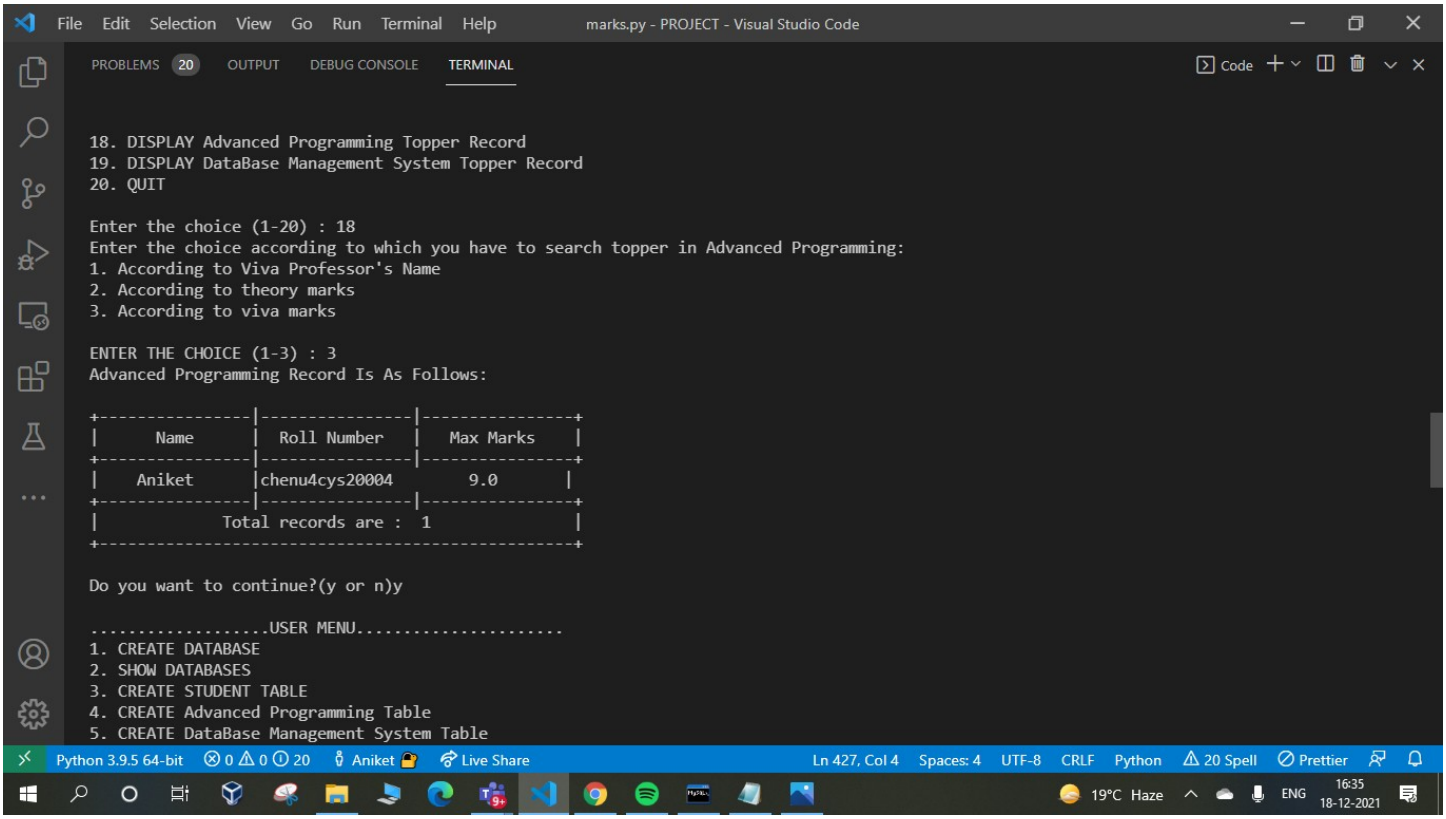
fes

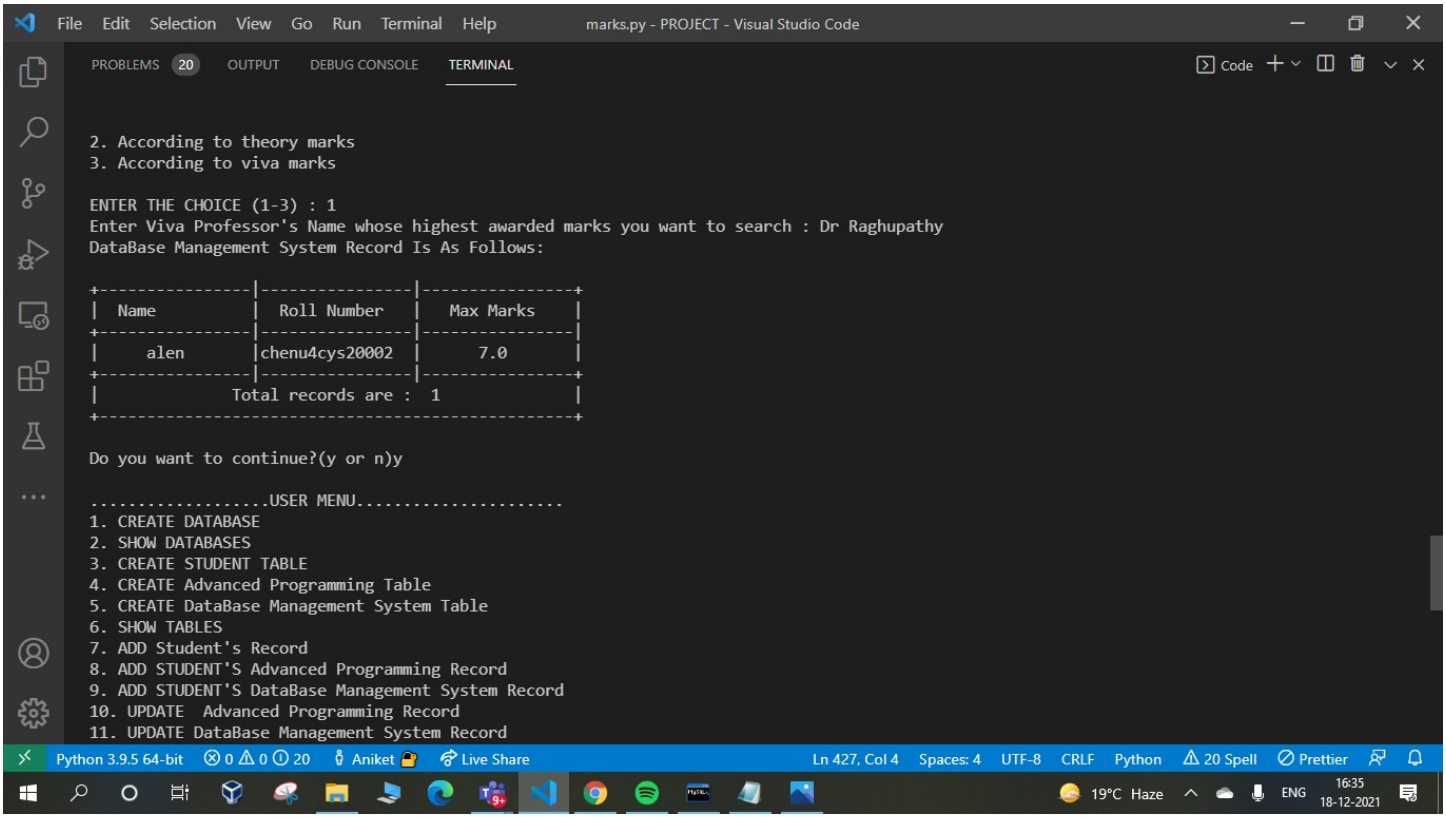

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 16
Successfully Connected
Advanced Programming Record is as follows:
Roll Number	Theory Marks	Viva Marks	Viva Professor
chenu4cys20001	7.0	8.0	Dr Vigneshwaran
chenu4cys20002	8.0	7.0	Dr Raghupathy
chenu4cys20003	6.0	7.0	Dr Raghupathy
chenu4cys20004	9.0	9.0	Dr Vigneshwaran
chenu4cys20005	7.0	8.0	Dr Aswiga
chenu4cys20006	7.0	7.0	Dr Sreedevi
chenu4cys20007	7.0	8.0	Dr Aswiga
chenu4cys20008	6.0	6.0	Dr Maheshwari
chenu4cys20009	7.0	8.0	Dr Vigneshawran
chenu4cys20010	6.0	8.0	Dr Sreedevi
Total no. of records are : 10
Do you want to continue?(y or n)
Python 3.9.5 64-bit Aniket Live Share
Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze ENG 16:30 18-12-2021

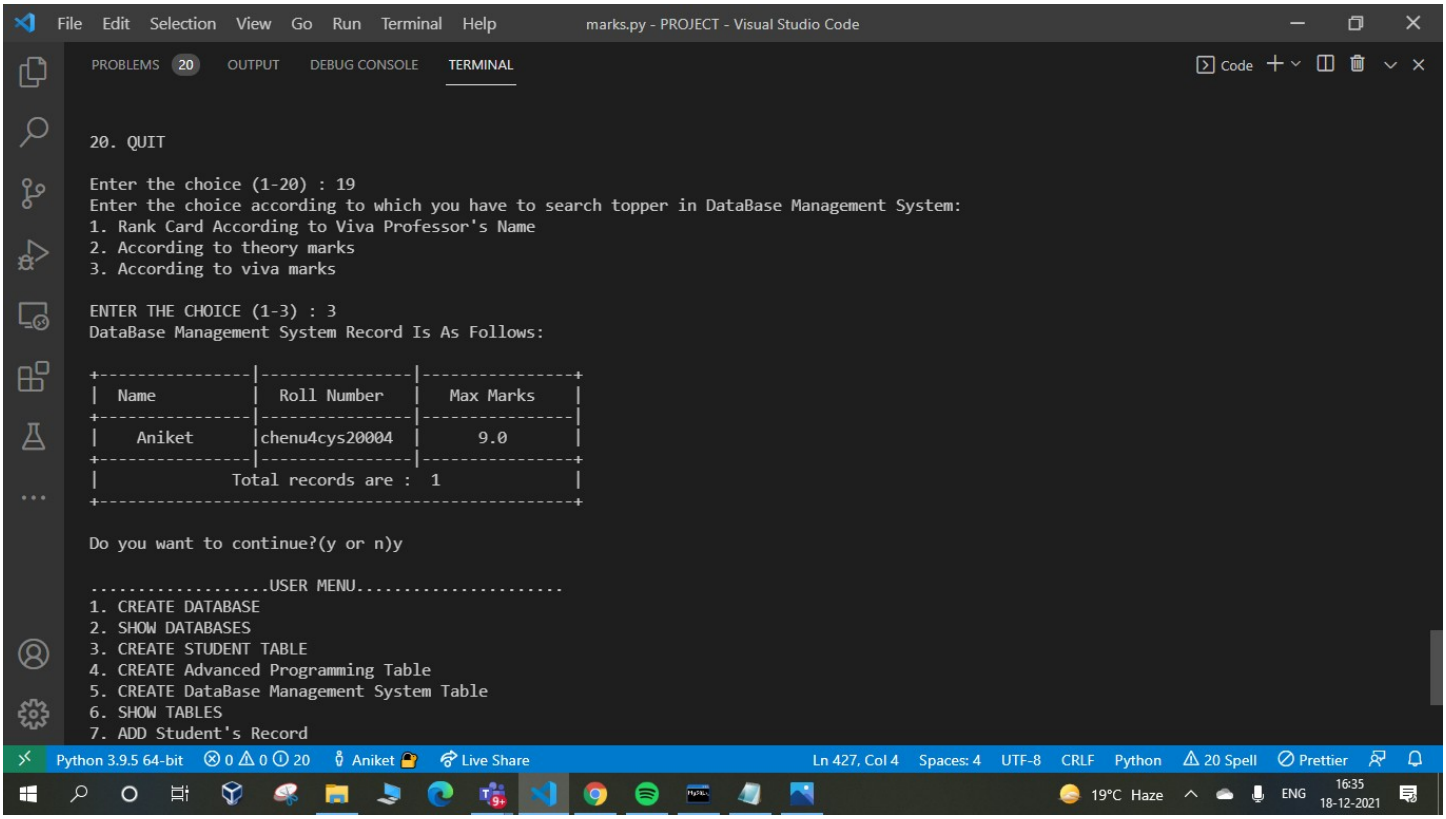

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 17
Successfully Connected
DataBase Management System Record Is As Follows:
Roll Number	Theory Marks	Viva Marks	Viva Professor
chenu4cys20001	6.0	7.0	Dr Vigneshawran
chenu4cys200010	6.0	7.0	Dr Vigneshwaran
chenu4cys20002	7.0	7.0	Dr Raghupathy
chenu4cys20003	7.0	6.0	Dr Vigneshawran
chenu4cys20004	9.0	9.0	Dr Vigneshwaran
chenu4cys20005	7.0	8.0	Dr Aswiga
chenu4cys20006	7.0	8.0	Dr Aswiga
chenu4cys20007	8.0	8.0	Dr Sreedevi
chenu4cys20008	6.0	6.0	Dr Sreedevi
chenu4cys20009	6.0	8.0	Dr Maheshwari
Total no. of records are : 10
Do you want to continue?(y or n)
Python 3.9.5 64-bit Aniket Live Share
Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Polluted air ENG 16:36 18-12-2021

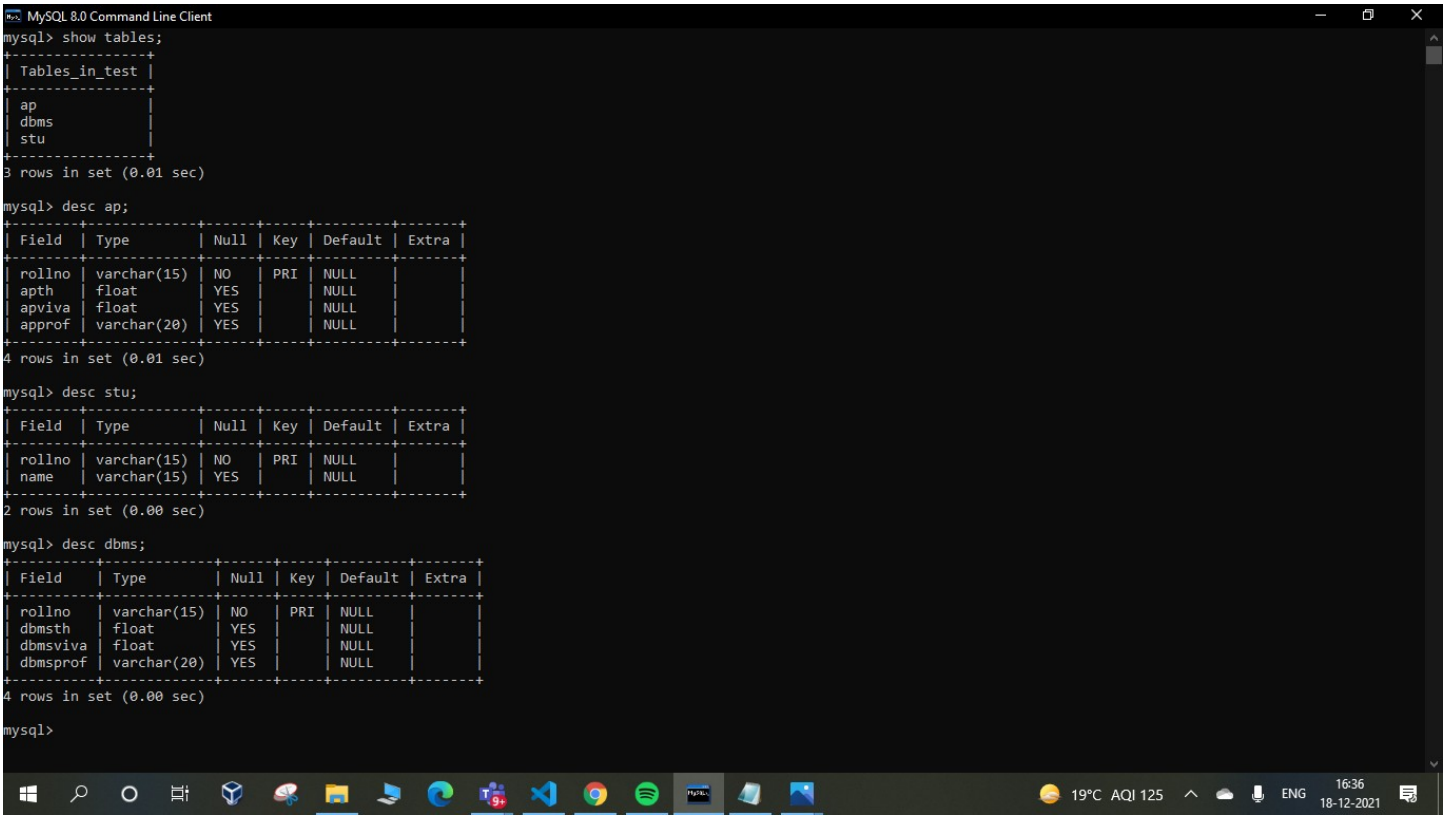

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
15. SEARCH DataBase Management System Record
16. DISPLAY Advanced Programming Record
17. DISPLAY DataBase Management System Record
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 18
Enter the choice according to which you have to search topper in Advanced Programming:
1. According to Viva Professor's Name
2. According to theory marks
3. According to viva marks
ENTER THE CHOICE (1-3) : 1
Enter Viva Professor's Name whose highest awarded marks you want to search : Dr Vigneshwaran
Advanced Programming Record Is As Follows:
Name	Roll Number	Max Marks
Aalap	chenu4cys20001	8.0
Aniket	chenu4cys20004	9.0
Total records are : 2
Do you want to continue?(y or n)y
.....USER MENU.....
Python 3.9.5 64-bit Aniket Live Share Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze ENG 16:34 18-12-2021

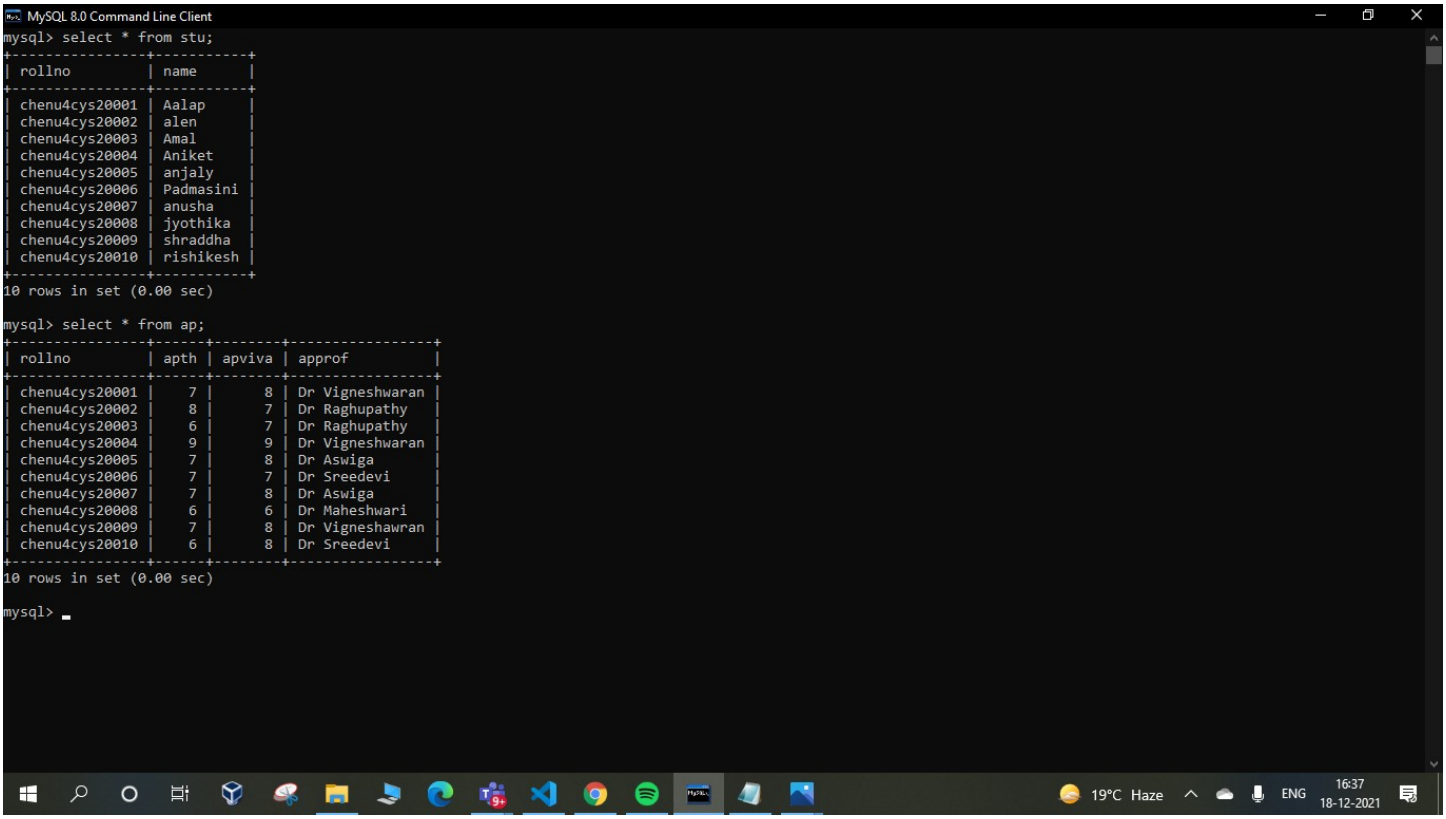

File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
Enter the choice (1-20) : 18
Enter the choice according to which you have to search topper in Advanced Programming:
1. According to Viva Professor's Name
2. According to theory marks
3. According to viva marks
ENTER THE CHOICE (1-3) : 2
Advanced Programming Record Is As Follows:
| Name | Roll Number | Max Marks |
| Aniket | chenu4cys20004 | 9.0 |
Total records are : 1
Do you want to continue?(y or n)y
.....USER MENU.....
1. CREATE DATABASE
2. SHOW DATABASES
3. CREATE STUDENT TABLE
4. CREATE Advanced Programming Table
5. CREATE DataBase Management System Table
6. SHOW TABLES
7. ADD Student's Record
8. ADD STUDENT'S Advanced Programming Record
Python 3.9.5 64-bit Aniket Live Share Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze ENG 16:35 18-12-2021


File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
18. DISPLAY Advanced Programming Topper Record
19. DISPLAY DataBase Management System Topper Record
20. QUIT
Enter the choice (1-20) : 18
Enter the choice according to which you have to search topper in Advanced Programming:
1. According to Viva Professor's Name
2. According to theory marks
3. According to viva marks
ENTER THE CHOICE (1-3) : 3
Advanced Programming Record Is As Follows:
| Name | Roll Number | Max Marks |
| Aniket | chenu4cys20004 | 9.0 |
Total records are : 1
Do you want to continue?(y or n)y
.....USER MENU.....
1. CREATE DATABASE
2. SHOW DATABASES
3. CREATE STUDENT TABLE
4. CREATE Advanced Programming Table
5. CREATE DataBase Management System Table
Python 3.9.5 64-bit
Aniket
Live Share
Ln 427, Col 4
Spaces: 4
UTF-8
CRLF
Python
20 Spell
Prettier
19°C Haze
ENG
16:35
18-12-2021


File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
2. According to theory marks
3. According to viva marks
ENTER THE CHOICE (1-3) : 1
Enter Viva Professor's Name whose highest awarded marks you want to search : Dr Raghupathy
DataBase Management System Record Is As Follows:
| Name | Roll Number | Max Marks |
| alen | chenu4cys20002 | 7.0 |
Total records are : 1
Do you want to continue?(y or n)y
.....USER MENU.....
1. CREATE DATABASE
2. SHOW DATABASES
3. CREATE STUDENT TABLE
4. CREATE Advanced Programming Table
5. CREATE DataBase Management System Table
6. SHOW TABLES
7. ADD Student's Record
8. ADD STUDENT'S Advanced Programming Record
9. ADD STUDENT'S DataBase Management System Record
10. UPDATE Advanced Programming Record
11. UPDATE DataBase Management System Record
Python 3.9.5 64-bit
Aniket
Live Share
Ln 427, Col 4
Spaces: 4
UTF-8
CRLF
Python
20 Spell
Prettier
19°C Haze
ENG
16:35
18-12-2021


File Edit Selection View Go Run Terminal Help
marks.py - PROJECT - Visual Studio Code
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL
Code
20. QUIT
Enter the choice (1-20) : 19
Enter the choice according to which you have to search topper in DataBase Management System:
1. Rank Card According to Viva Professor's Name
2. According to theory marks
3. According to viva marks
ENTER THE CHOICE (1-3) : 3
DataBase Management System Record Is As Follows:
| Name | Roll Number | Max Marks |
| Aniket | chenu4cys20004 | 9.0 |
Total records are : 1
Do you want to continue?(y or n)y
.....USER MENU.....
1. CREATE DATABASE
2. SHOW DATABASES
3. CREATE STUDENT TABLE
4. CREATE Advanced Programming Table
5. CREATE DataBase Management System Table
6. SHOW TABLES
7. ADD Student's Record
Python 3.9.5 64-bit 0 0 20 Aniket Live Share
Ln 427, Col 4 Spaces: 4 UTF-8 CRLF Python 20 Spell Prettier
19°C Haze ENG 16:35 18-12-2021


MySQL 8.0 Command Line Client
mysql> show tables;
| Tables_in_test |
| ap |
| dbms |
| stu |
3 rows in set (0.01 sec)
mysql> desc ap;
Field	Type	Null	Key	Default	Extra
rollno	varchar(15)	NO	PRI	NULL	
apth	float	YES		NULL	
apviva	float	YES		NULL	
approf	varchar(20)	YES		NULL	
4 rows in set (0.01 sec)
mysql> desc stu;
Field	Type	Null	Key	Default	Extra
rollno	varchar(15)	NO	PRI	NULL	
name	varchar(15)	YES		NULL	
2 rows in set (0.00 sec)
mysql> desc dbms;
Field	Type	Null	Key	Default	Extra
rollno	varchar(15)	NO	PRI	NULL	
dbmsth	float	YES		NULL	
dbmsviva	float	YES		NULL	
dbmsprof	varchar(20)	YES		NULL	
4 rows in set (0.00 sec)
mysql>
19°C AQI 125 ENG 16:36 18-12-2021



REFERENCES

- https://stackoverflow.com/questions/39149243/how-do-i-connect-to-a-sql-server-database-with-python
- https://www.tutorialspoint.com/python/python_database_access.htm
- https://www.w3schools.in/python-tutorial/database-connection/
- https://creately.com/diagram-type/database-design
- https://github.com/topics/dbms?l=python
- https://cloud.smartdraw.com/editor.aspx?templateId=da34e096-b9cb-4d56-a0ce-d9bdef138714&flags=128#depoId=31723575&credID=-36858967
- https://stackoverflow.com/questions/2655748/writing-a-dbms-in-python
- https://www.codewithc.com/pharmacy-management-system-c-mysql/#google_vignette
- https://docs.python.org/3/library/sqlite3.html
- https://www.freecodecamp.org/news/connect-python-with-sql/