Batch: A3	**Roll No.:** 1911034
Experiment / assignment / tutorial No.________	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Demonstrating Database Connectivity in Python

AIM: To understand Database Connectivity in Python

Expected OUTCOME of Experiment:

CO - 3

Books/ Journals/ Websites referred:

1. Geeksforgeeks
2. TutorialsPoint
3. W3Schools
4. Ma'am's ppt

Pre Lab/ Prior Concepts:

Creating the connection

To create a connection between the MySQL database and the python application, the connect() method of mysql.connector module is used.

Pass the database details like HostName, username, and the database password in the method call. The method returns the connection object.

The syntax to use the connect() is given below.

1. Connection-Object= mysql.connector.connect(host = <host-name> , user = <username> , passwd = <password>)

Consider the following example.

Example

```
import mysql.connector

#Create the connection object
myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "google")

#printing the connection object
print(myconn)
```

Output:

```
<mysql.connector.connection.MySQLConnection object at 0x7fb142edd780>
```

Here, we must notice that we can specify the database name in the connect() method if we want to connect to a specific database.

Example

```
import mysql.connector

#Create the connection object
myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "google", database = "mydb")

#printing the connection object
print(myconn)
```

Output:

```
<mysql.connector.connection.MySQLConnection object at 0x7ff64aa3d7b8>
```

Creating a cursor object

The cursor object can be defined as an abstraction specified in the Python DB-API 2.0. It facilitates us to have multiple separate working environments through the same connection to the database. We can create the cursor object by calling the 'cursor' function of the connection object. The cursor object is an important aspect of executing queries to the databases.

The syntax to create the cursor object is given below.

```
<my_cur>  = conn.cursor()
```

Example

```
import mysql.connector
#Create the connection object
myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "google", database = "mydb")

#printing the connection object
print(myconn)

#creating the cursor object
cur = myconn.cursor()

print(cur)
```

Output:

```
<mysql.connector.connection.MySQLConnection object at 0x7faa17a15748>
MySQLCursor: (Nothing executed yet)
```

Problem statement:

Write a program to accept the users input for username and password to register. Write a program to check the validity of Username and password input by users.

Following are the criteria for checking the password:

1. At least 1 letter between [a-z]

2. At least 1 number between [0-9]

1. At least 1 letter between [A-Z]

3. At least 1 character from [$#@]

4. Minimum length of transaction password: 8

5. Maximum length of transaction password: 15

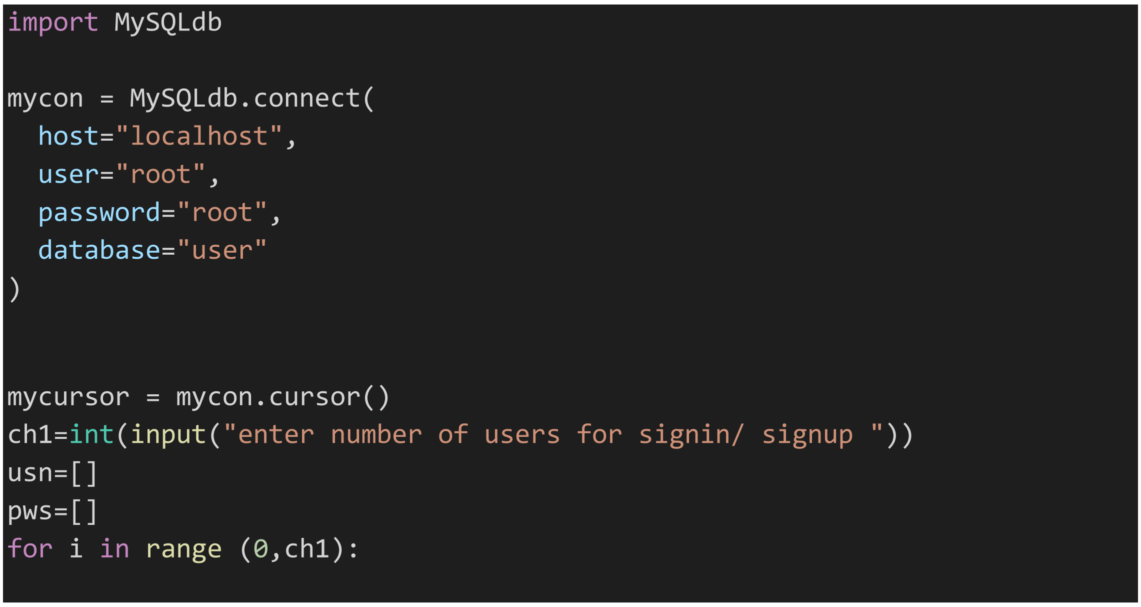
Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma.

Note: Unique username, valid password based on rules.

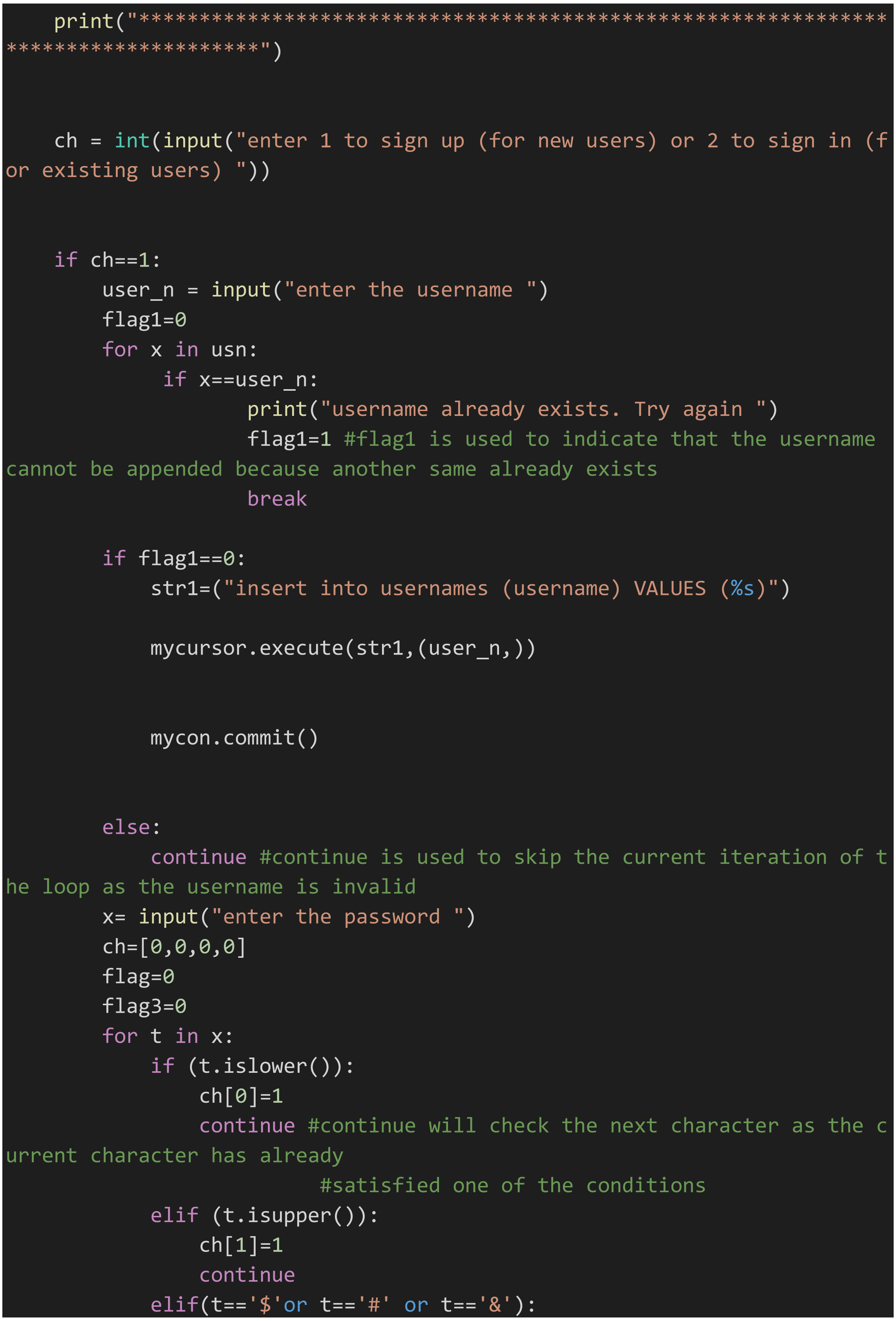
signup

Modify the above experiment to demonstrate database connectivity.

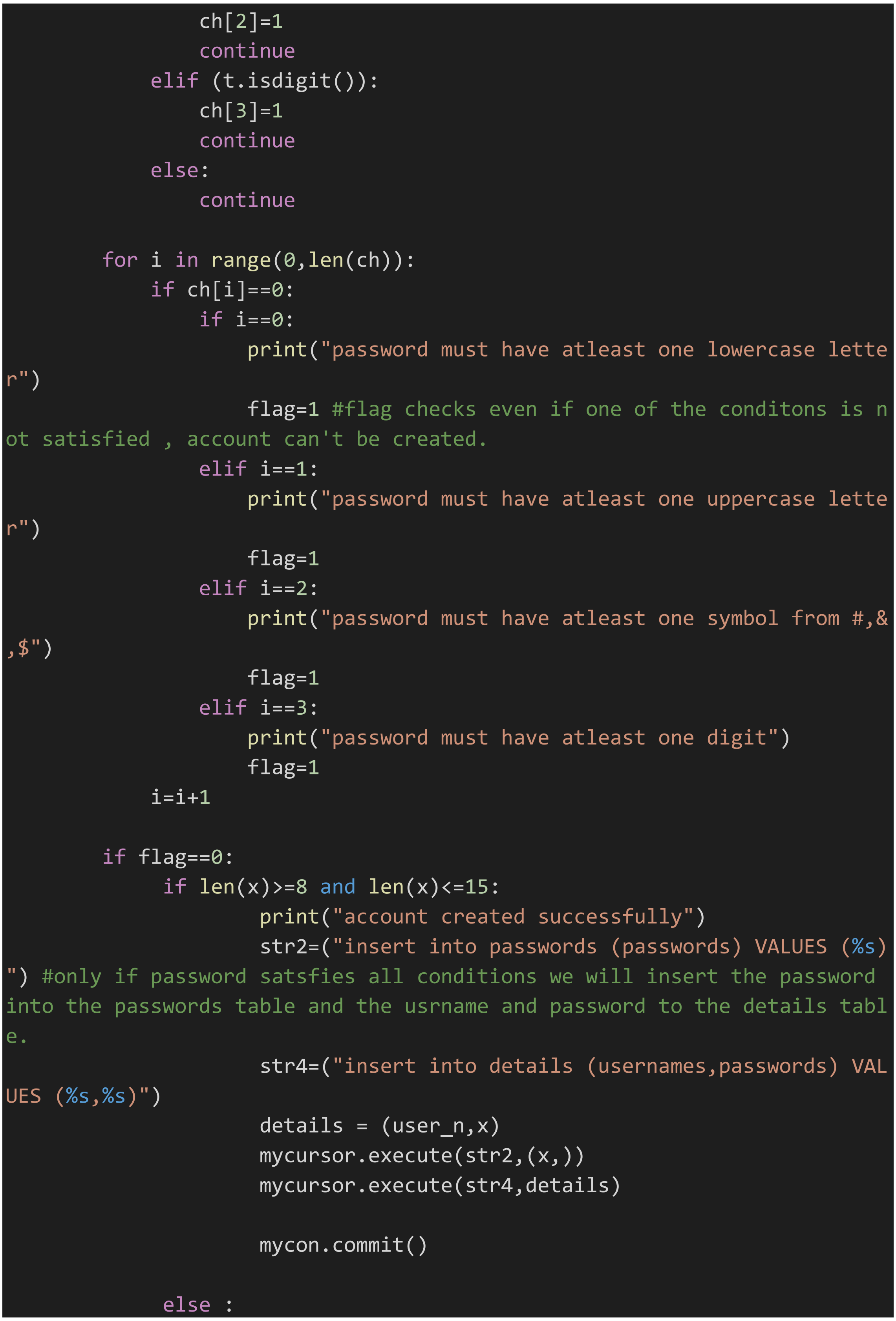
Program (with comments) (Write comments to your program to explain the implementation)



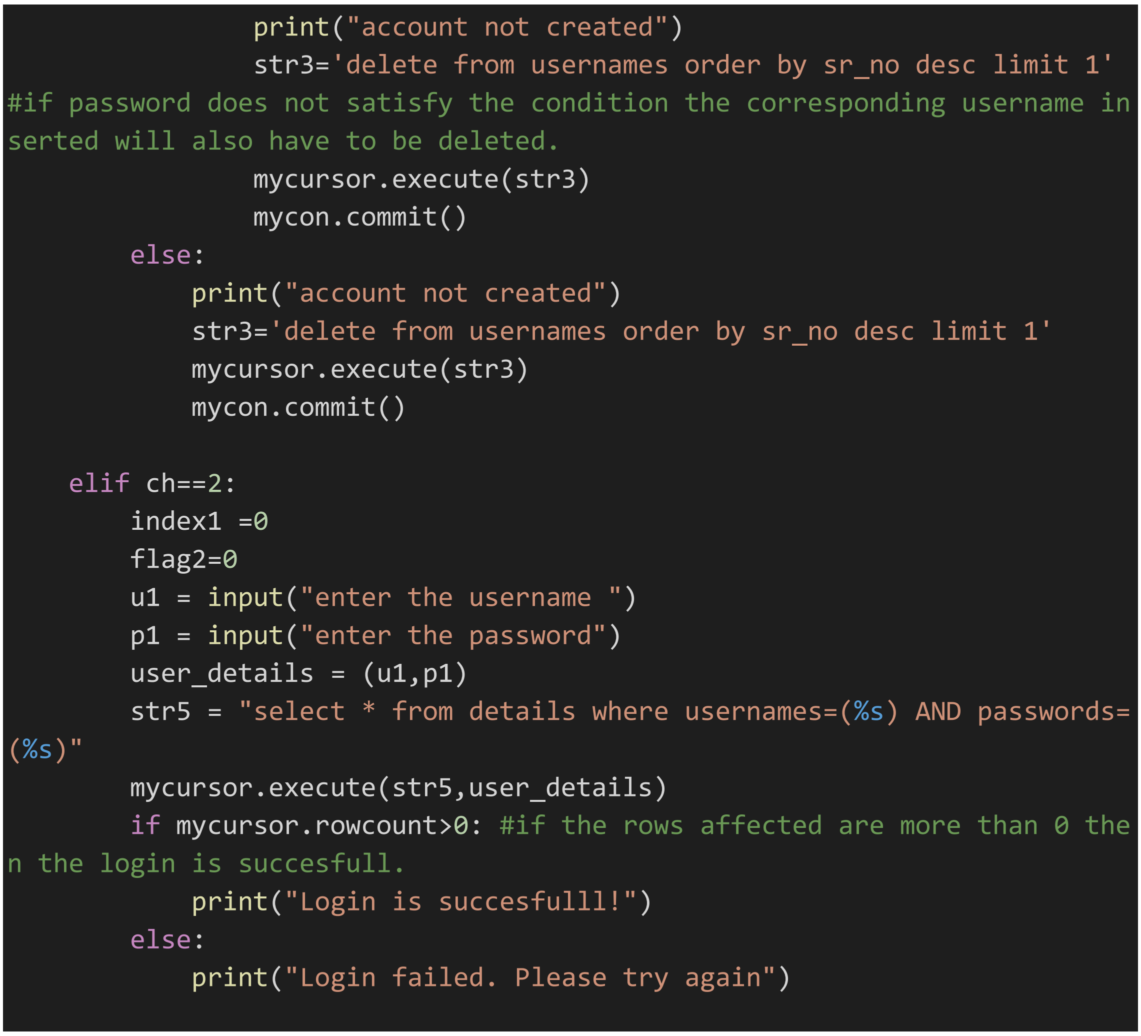
```
import MySQLdb

mycon = MySQLdb.connect(
  host="localhost",
  user="root",
  password="root",
  database="user"
)


mycursor = mycon.cursor()
ch1=int(input("enter number of users for signin/ signup "))
usn=[]
pws=[]
for i in range (0,ch1):
```



```
    print("*************************************************************
*********************")


    ch = int(input("enter 1 to sign up (for new users) or 2 to sign in (f
or existing users) "))


    if ch==1:
        user_n = input("enter the username ")
        flag1=0
        for x in usn:
             if x==user_n:
                    print("username already exists. Try again ")
                    flag1=1 #flag1 is used to indicate that the username
cannot be appended because another same already exists
                    break

        if flag1==0:
            str1=("insert into usernames (username) VALUES (%s)")

            mycursor.execute(str1,(user_n,))


            mycon.commit()


        else:
            continue #continue is used to skip the current iteration of t
he loop as the username is invalid
        x= input("enter the password ")
        ch=[0,0,0,0]
        flag=0
        flag3=0
        for t in x:
            if (t.islower()):
                ch[0]=1
                continue #continue will check the next character as the c
urrent character has already
                          #satisfied one of the conditions
            elif (t.isupper()):
                ch[1]=1
                continue
            elif(t=='$'or t=='#' or t=='&'):
```



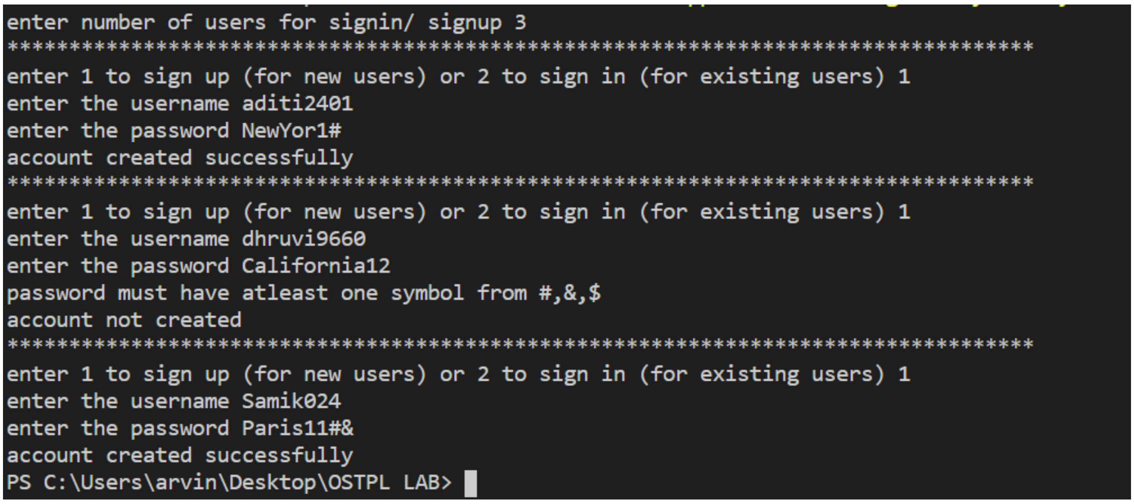
```
                ch[2]=1
                continue
            elif (t.isdigit()):
                ch[3]=1
                continue
            else:
                continue

        for i in range(0,len(ch)):
            if ch[i]==0:
                if i==0:
                    print("password must have atleast one lowercase lette
r")
                    flag=1 #flag checks even if one of the conditons is n
ot satisfied , account can't be created.
                elif i==1:
                    print("password must have atleast one uppercase lette
r")
                    flag=1
                elif i==2:
                    print("password must have atleast one symbol from #,&
,$")
                    flag=1
                elif i==3:
                    print("password must have atleast one digit")
                    flag=1
            i=i+1

        if flag==0:
             if len(x)>=8 and len(x)<=15:
                     print("account created successfully")
                     str2=("insert into passwords (passwords) VALUES (%s)
") #only if password satsfies all conditions we will insert the password
into the passwords table and the usrname and password to the details tabl
e.
                     str4=("insert into details (usernames,passwords) VAL
UES (%s,%s)")
                     details = (user_n,x)
                     mycursor.execute(str2,(x,))
                     mycursor.execute(str4,details)

                     mycon.commit()

             else :
```



```
                print("account not created")
                str3='delete from usernames order by sr_no desc limit 1'
#if password does not satisfy the condition the corresponding username in
serted will also have to be deleted.
                mycursor.execute(str3)
                mycon.commit()
        else:
            print("account not created")
            str3='delete from usernames order by sr_no desc limit 1'
            mycursor.execute(str3)
            mycon.commit()

    elif ch==2:
        index1 =0
        flag2=0
        u1 = input("enter the username ")
        p1 = input("enter the password")
        user_details = (u1,p1)
        str5 = "select * from details where usernames=(%s) AND passwords=
(%s)"
        mycursor.execute(str5,user_details)
        if mycursor.rowcount>0: #if the rows affected are more than 0 the
n the login is succesfull.
            print("Login is succesfulll!")
        else:
            print("Login failed. Please try again")
```

Output:

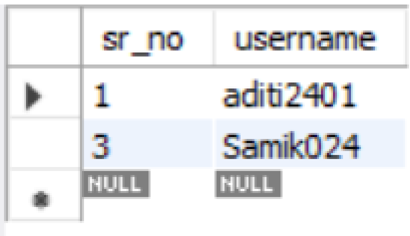
First we ask for three queries from which only two are successfully inserted as one of them does not satisfy all the conditions for the password:



```
enter number of users for signin/ signup 3
**********************************************************************************
enter 1 to sign up (for new users) or 2 to sign in (for existing users) 1
enter the username aditi2401
enter the password NewYor1#
account created successfully
**********************************************************************************
enter 1 to sign up (for new users) or 2 to sign in (for existing users) 1
enter the username dhruvi9660
enter the password California12
password must have atleast one symbol from #,&,$
account not created
**********************************************************************************
enter 1 to sign up (for new users) or 2 to sign in (for existing users) 1
enter the username Samik024
enter the password Paris11#&
account created successfully
PS C:\Users\arvin\Desktop\OSTPL LAB> █
```

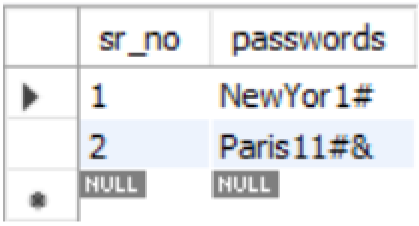
The database now looks like this:

Usernames table:



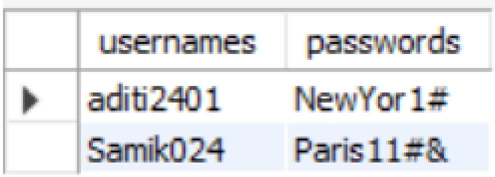
	sr_no	username
▶	1	aditi2401
	3	Samik024
*	NULL	NULL

Passwords table:



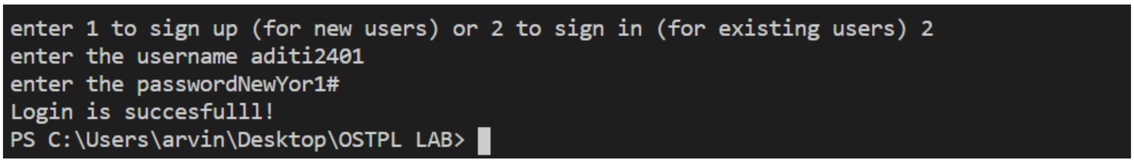
	sr_no	passwords
▶	1	NewYor1#
	2	Paris11#&
*	NULL	NULL

Details table:



	usernames	passwords
▶	aditi2401	NewYor1#
	Samik024	Paris11#&

Implementing Login functionality:



```
enter 1 to sign up (for new users) or 2 to sign in (for existing users) 2
enter the username aditi2401
enter the passwordNewYor1#
Login is succesfulll!
PS C:\Users\arvin\Desktop\OSTPL LAB> 
```

Conclusion: In this experiment the concept of database connectivity in python was successfully learnt and implemented**.**

Date: _____________ **Signature of faculty in-charge**