Assignment 2

Team ID	PNT2022TMID14112
Project Name	Plasma Donor Application

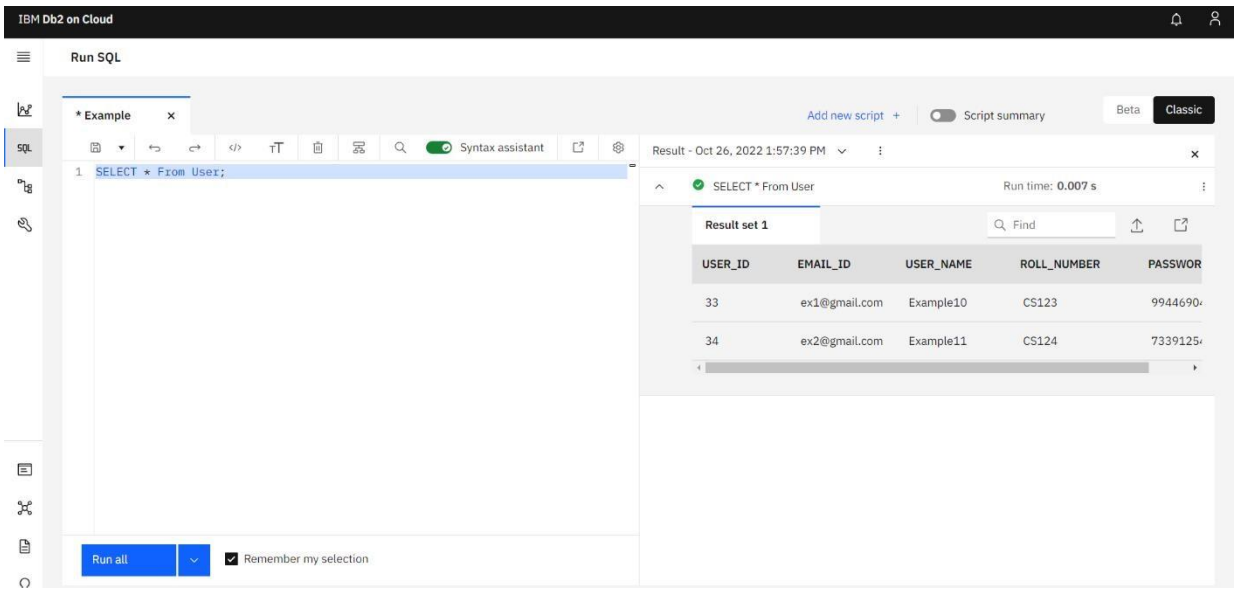
Question:

1. Create User table with user with email, username, roll number, password.
2. Perform UPDATE, DELETE Queries with user table
3. Connect python code to db2.
4. Create a flask app with registration page, login page and welcome page. By default, load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid show the welcome page.

1.Create User table with user with email, username, roll number, password.

Solution:

```
CREATE TABLE USER( USER_ID INT GENERATED BY DEFAULT AS IDENTITY NOT NULL, EMAIL_ID VARCHAR(150) NOT NULL, USER_NAME VARCHAR(255) NOT NULL, ROLL_NUMBER VARCHAR(100) NOT NULL, PASSWORD VARCHAR(100) NOT NULL, PRIMARY KEY (USER_ID) ); INSERT INTO USER(EMAIL_ID, USER_NAME,ROLL_NUMBER,PASSWORD) VALUES('ex1@gmail.com','Example10','CS123','9944690406');

INSERT INTO USER(EMAIL_ID, USER_NAME,ROLL_NUMBER,PASSWORD) VALUES('ex2@gmail.com','Example11','CS124','7339125450');
```

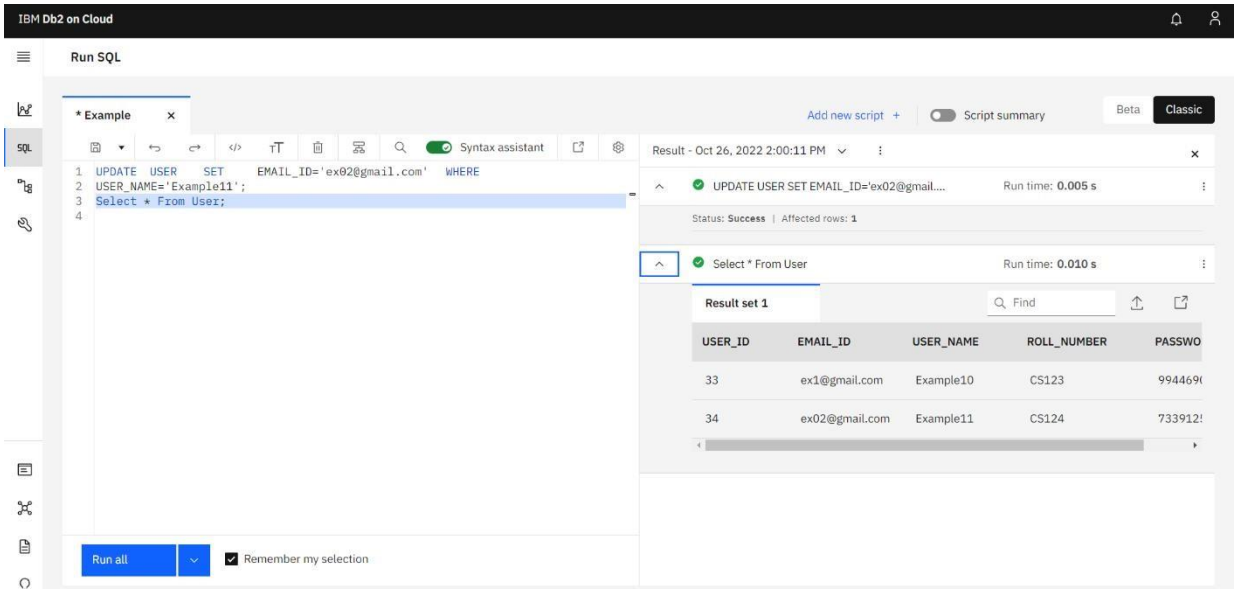


2.Perform UPDATE, DELETE Queries with user table.

Solution:

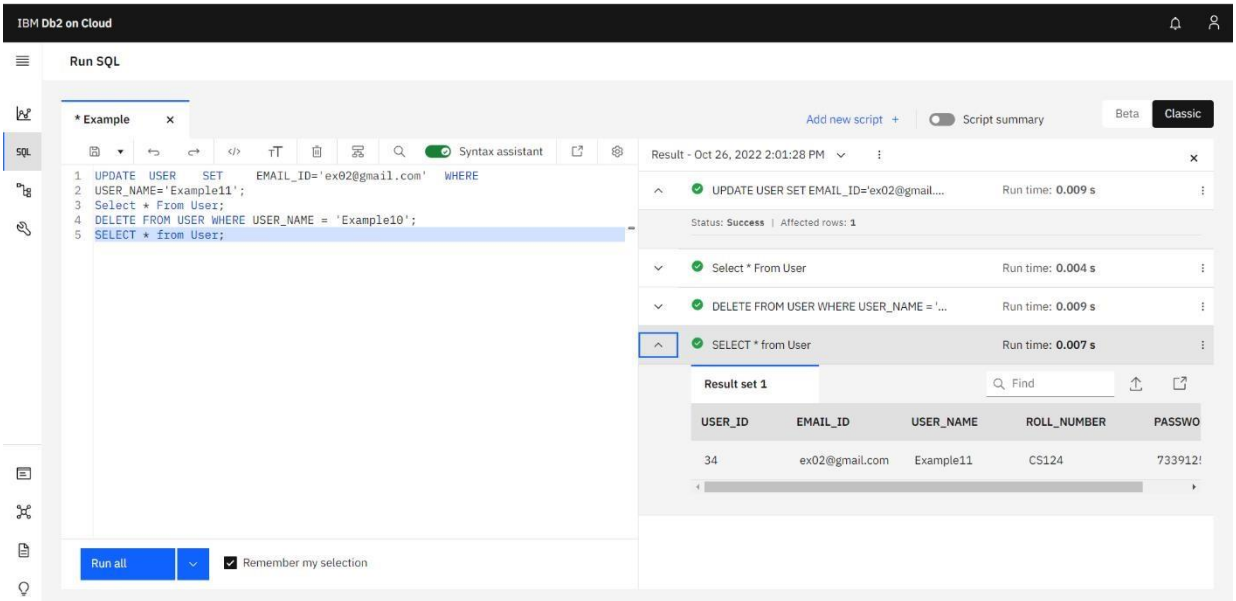
UPDATE:

```
UPDATE USER SET EMAIL_ID='ex02@gmail.com' WHERE USER_NAME='Example11';
```



DELETE:

```
DELETE FROM USER WHERE USER_NAME = 'Example10';
```



3. Connect python code to db2.

Solution:

def Connection():

try:

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dkf92060;PWD=aUIIMe5Ip1OpH15a ", "", "") print ("Database Connected Successfully !") return conn except:

print ("Unable to connect: ", ibm_db.conn_errormsg())

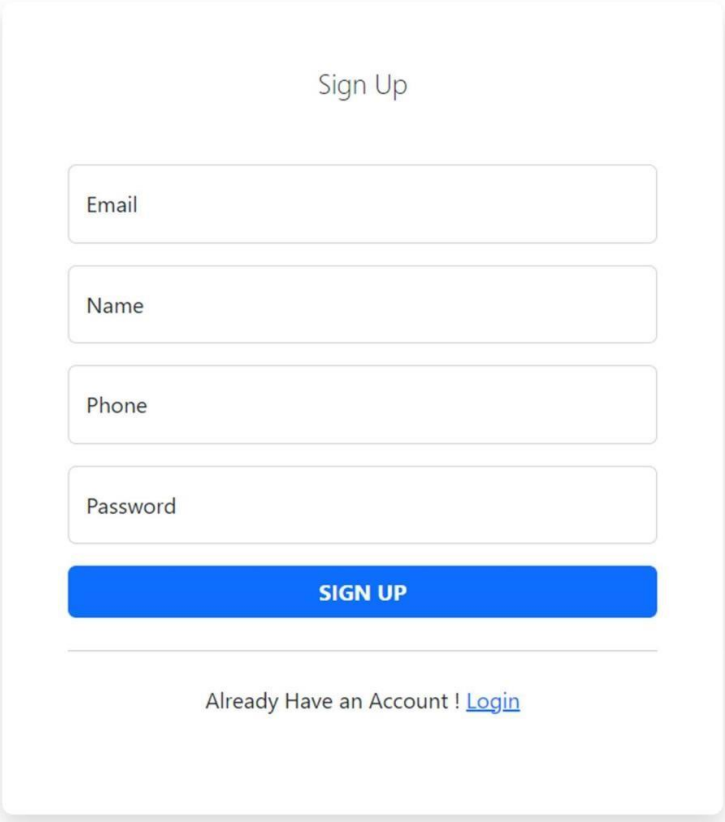
4. Create a flask app with registration page, login page and welcome page. By default, load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid show the welcome page.

```
 import ibm_db

def Connection():
    try:
        conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dkf92060;PWD=aUIIMe5Ip1OpH15
a", "", "")
        print ("Database Connected Successfully !")
        return conn
    except:
        print ("Unable to connect: ", ibm_db.conn_errormsg())
def Create(email,name,phone,password,conn):

    columns = '"UNAME","UEMAIL","UPHONE","UPASSWORD"'
    val = "'"+name+"','"+email+"','"+phone+"','"+password+"'"
    sql = 'Insert into DKF92060.USER(' + columns + ') values('+val+')'
    try:
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(stmt)
        print ("added :-)")
        return 1
    except:
        print("Error While Adding the User ! ")
        return 0
```

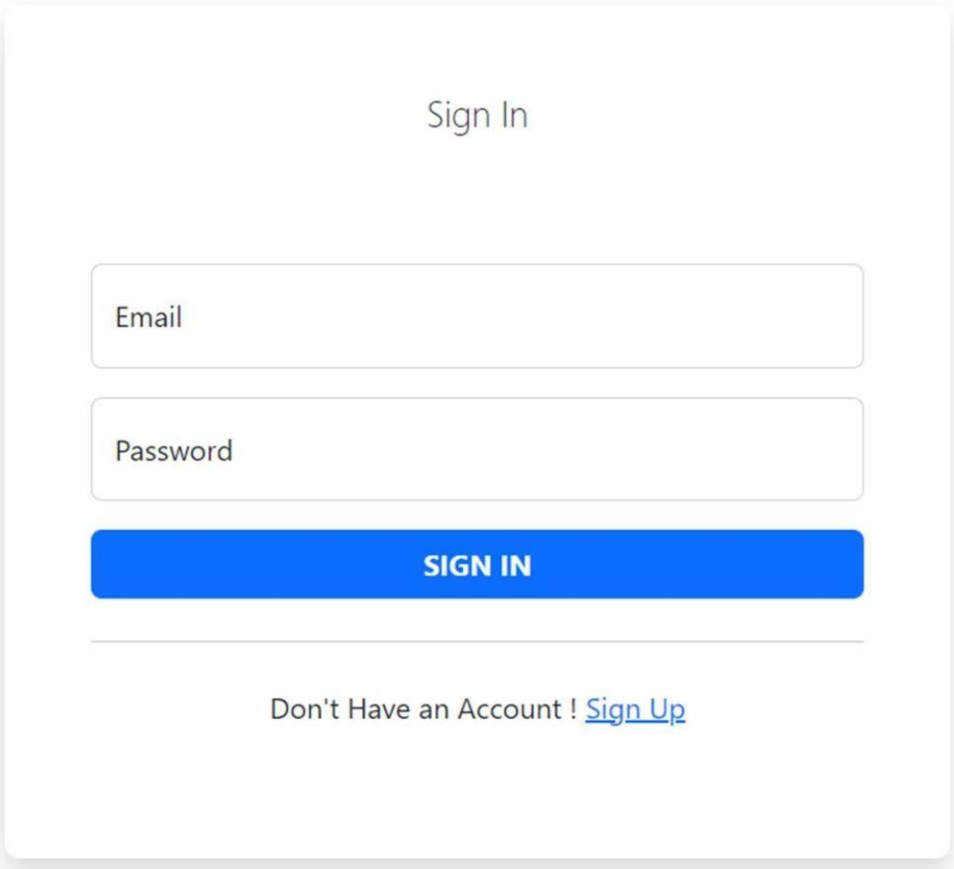
```
def Signin(email,password,conn):

    sql = "SELECT * FROM DKF92060.USER"
    try:
        result = ibm_db.exec_immediate(conn,sql)
        tuple = ibm_db.fetch_tuple(result)
        while tuple != False:
            if str(tuple[2]) == email and str(tuple[4]) == password:
                res = [str(tuple[1]),str(tuple[2]),str(tuple[3])]
                return res
            tuple = ibm_db.fetch_tuple(result)
        print("Fetch Success :-)")
        return 0
    except:
        print("fetch not found !")
        return 0
```

Output:

- Signup Page:



- Login Page:



- When Email or password is Invalid:

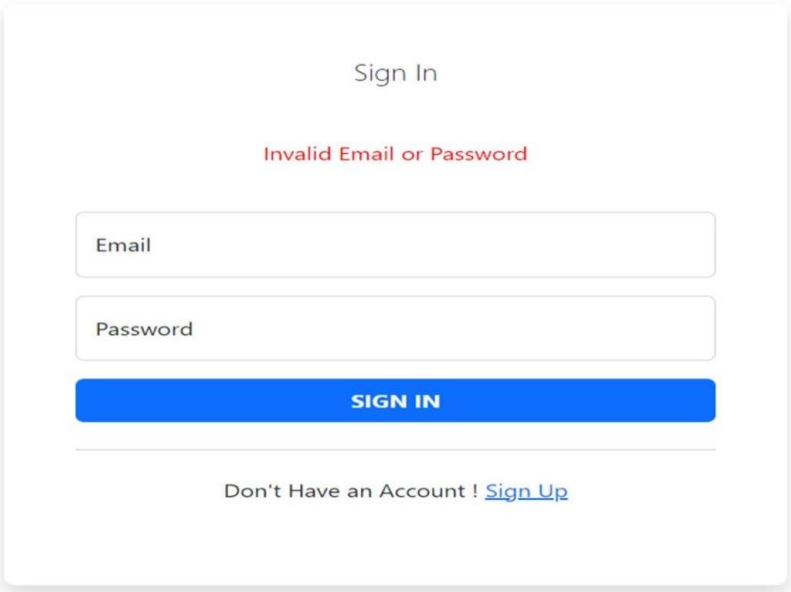


Table:

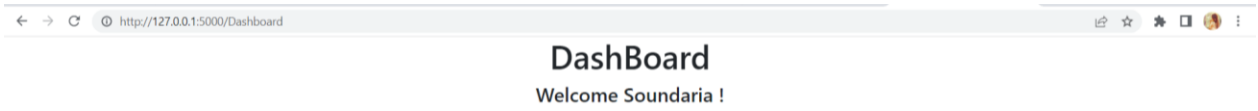
DKF92060.USER

Back

Export to CSV

USER_ID	UNAME	UEMAIL	UPHONE	UPASSWORD
1	Soundaria	711319CS155@smartinternz.com	7339125450	8763
2	Soundaria	example@gmail.com	7339125450	0987654321

Home Page:



DashBoard

Welcome Soundaria !