

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');
```

The screenshot shows the IBM Db2 on Cloud console interface. The top bar indicates 'IBM Db2 on Cloud'. The main area is titled 'Run SQL'. On the left, there's a sidebar with icons for SQL, Data, and other tools. The central editor shows a SQL query: '1. SELECT * From User;'. The right panel displays the results of the query, titled 'Result - Oct 26, 2022 1:57:39 PM'. It shows a table with 5 columns: USER_ID, EMAIL_ID, USER_NAME, ROLL_NUMBER, and PASSWORD. The results are as follows:

USER_ID	EMAIL_ID	USER_NAME	ROLL_NUMBER	PASSWOR
33	ex1@gmail.com	Example10	CS123	9944690406
34	ex2@gmail.com	Example11	CS124	7339125450

At the bottom, there are buttons for 'Run all' and a checkbox for 'Remember my selection'.

2.Perform UPDATE, DELETE Queries with user table.

Solution:

UPDATE:

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

The screenshot shows the IBM Db2 on Cloud Run SQL interface. The SQL editor contains the following queries:

```
1. UPDATE USER SET EMAIL_ID='ex02@gmail.com' WHERE  
2. USER_NAME='Example11';  
3. Select * From User;  
4.
```

The results pane shows the execution of the UPDATE query and the subsequent SELECT query. The UPDATE query was successful, affecting 1 row. The SELECT query returned the following data:

USER_ID	EMAIL_ID	USER_NAME	ROLL_NUMBER	PASSWO
33	ex1@gmail.com	Example10	CS123	9944690
34	ex02@gmail.com	Example11	CS124	7339121

DELETE:

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

The screenshot shows the IBM Db2 on Cloud Run SQL interface. The SQL editor contains the following queries:

```
1. UPDATE USER SET EMAIL_ID='ex02@gmail.com' WHERE  
2. USER_NAME='Example11';  
3. Select * From User;  
4. DELETE FROM USER WHERE USER_NAME = 'Example10';  
5. SELECT * from User;
```

The results pane shows the execution of the UPDATE query, the DELETE query, and the subsequent SELECT query. The UPDATE query was successful, affecting 1 row. The DELETE query was also successful, affecting 1 row. The SELECT query returned the following data:

USER_ID	EMAIL_ID	USER_NAME	ROLL_NUMBER	PASSWO
34	ex02@gmail.com	Example11	CS124	7339121

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=aUIIME5Ip10pH15a", "", "")  
        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-4bb0-85b9ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dkf92060;PWD=aUIIME5Ip10pH15a", "", "")  
        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:

Sign Up

Already Have an Account ! [Login](#)

- Login Page:

Sign In

Email

Password

SIGN IN

Don't Have an Account ! [Sign Up](#)

- When Email or password is Invalid:

Sign In

Invalid Email or Password

Email

Password

SIGN IN

Don't Have an Account ! [Sign Up](#)

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:

http://127.0.0.1:5000/Dashboard

DashBoard

Welcome Soundaria !