Assignment -2

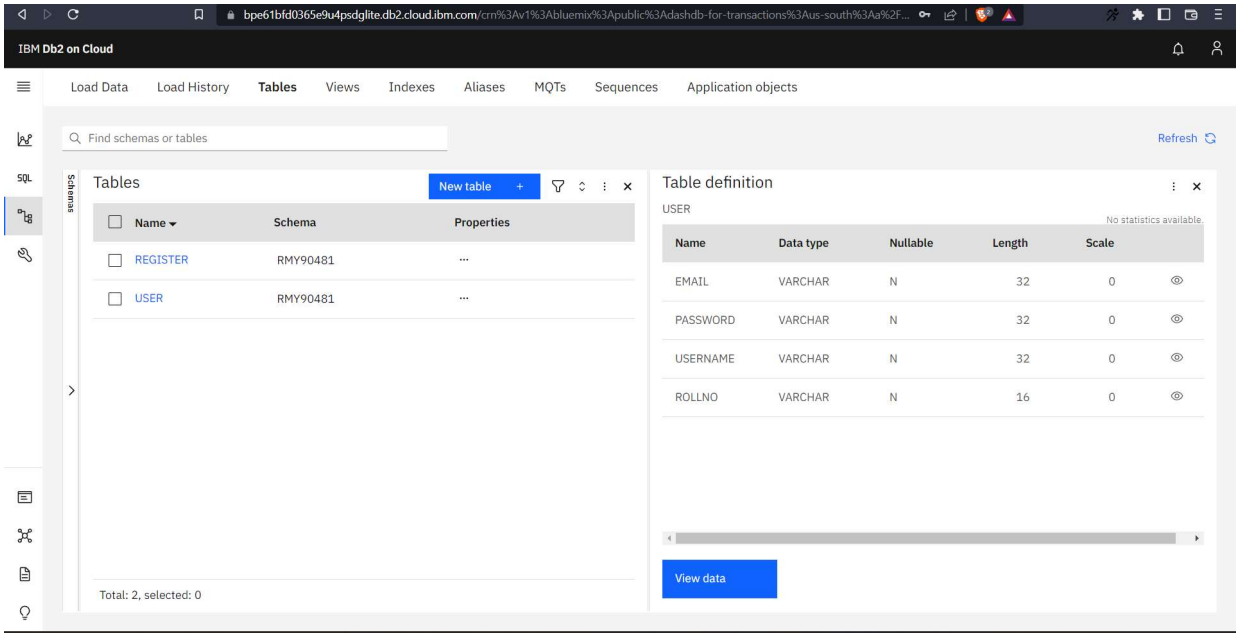
Python Programming

Assignment Date	23 September 2022
Student Name	Sudhirkumar S
Student Roll Number	2019115110
Maximum Marks	2 Marks

Question 1:

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

Solution 1:



Question 2:

Perform UPDATE, DELETE Queries with user table

Solution 2:

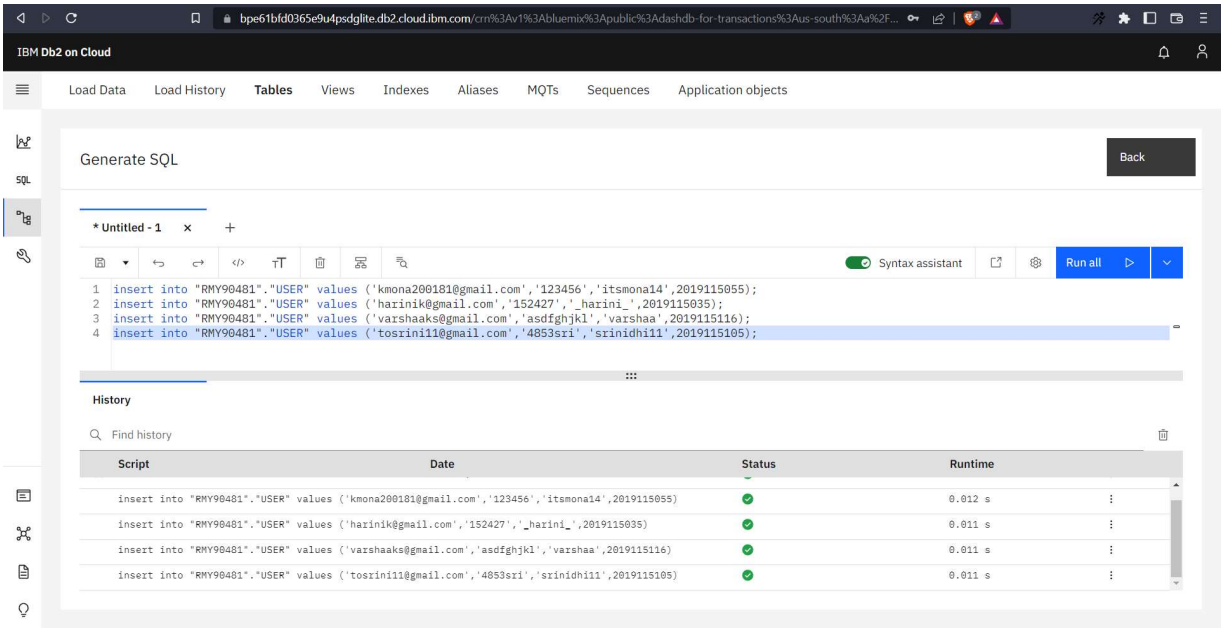
INSERT

insert into USER values ('kmona200181@gmail.com','123456','itsmona14',2019115055);

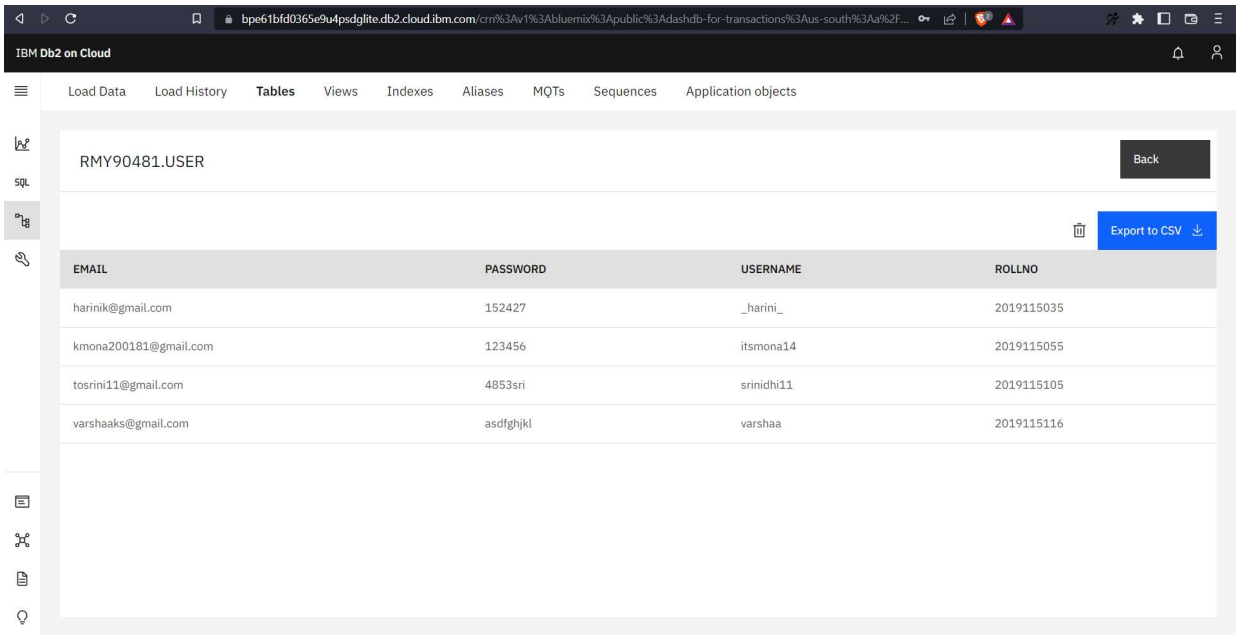
insert into USER values ('harinik@gmail.com','152427','_harini_',2019115035);

insert into USER values ('varshaaks@gmail.com','asdfghjkl','varshaa',2019115116);

insert into USER values ('tosrini11@gmail.com','4853sri','srinidhi11',2019115105);

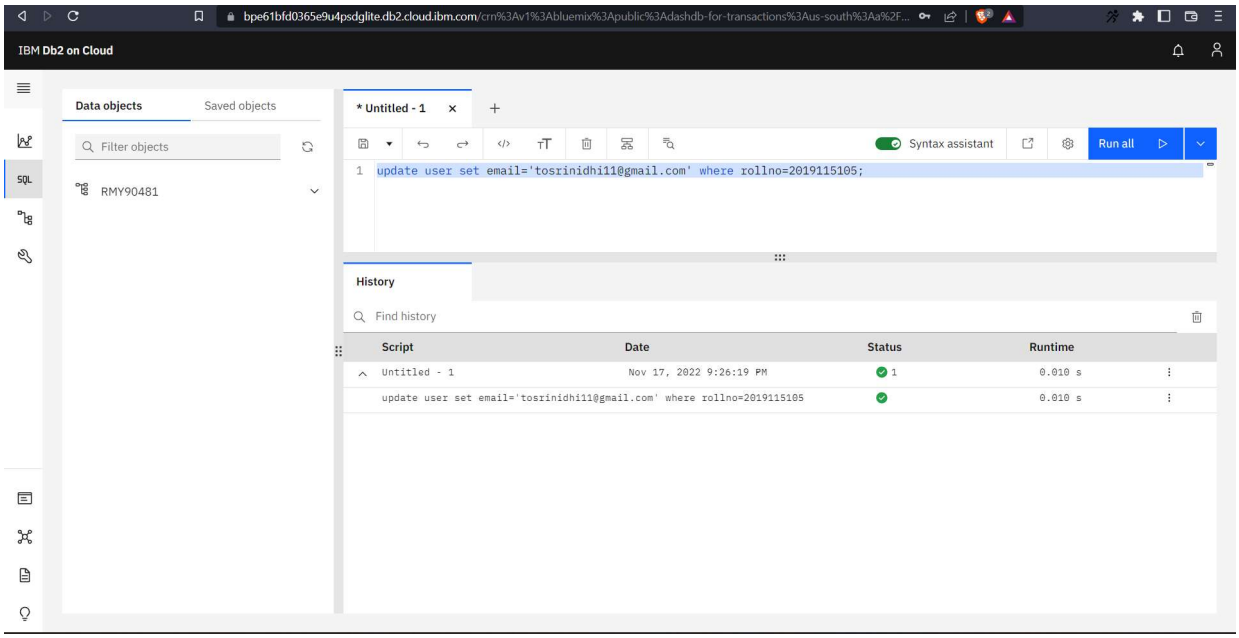


Output:

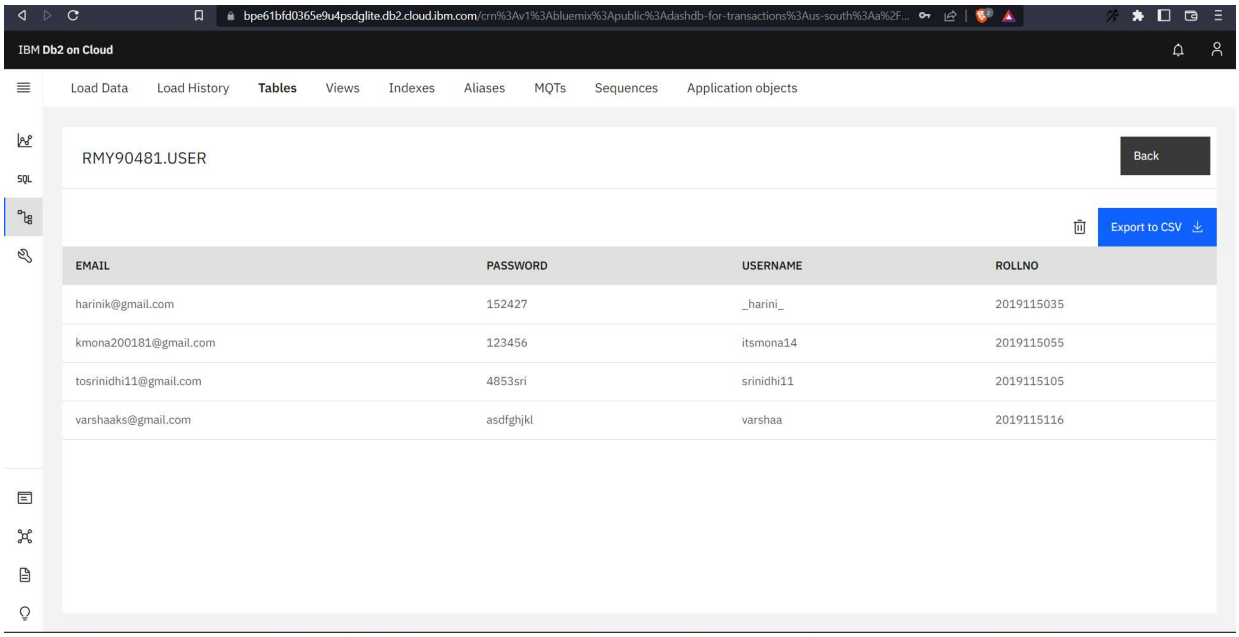


UPDATE

update user set email='tosrinidhi11@gmail.com' where rollno=2019115105

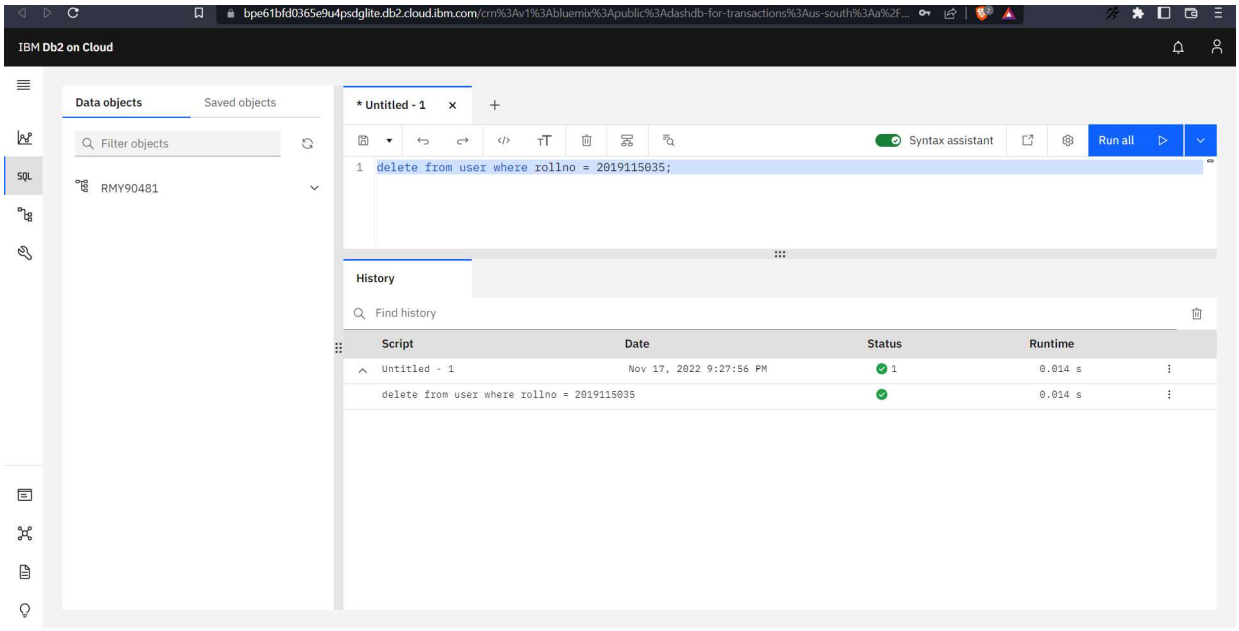

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F...
IBM Db2 on Cloud
Data objects
Saved objects
Filter objects
RMY90481
* Untitled - 1
Syntax assistant
Run all
1 update user set email='tosrinidhi11@gmail.com' where rollno=2019115105;
History
Find history
Script
Date
Status
Runtime
Untitled - 1
Nov 17, 2022 9:26:19 PM
1
0.010 s
update user set email='tosrinidhi11@gmail.com' where rollno=2019115105
0.010 s

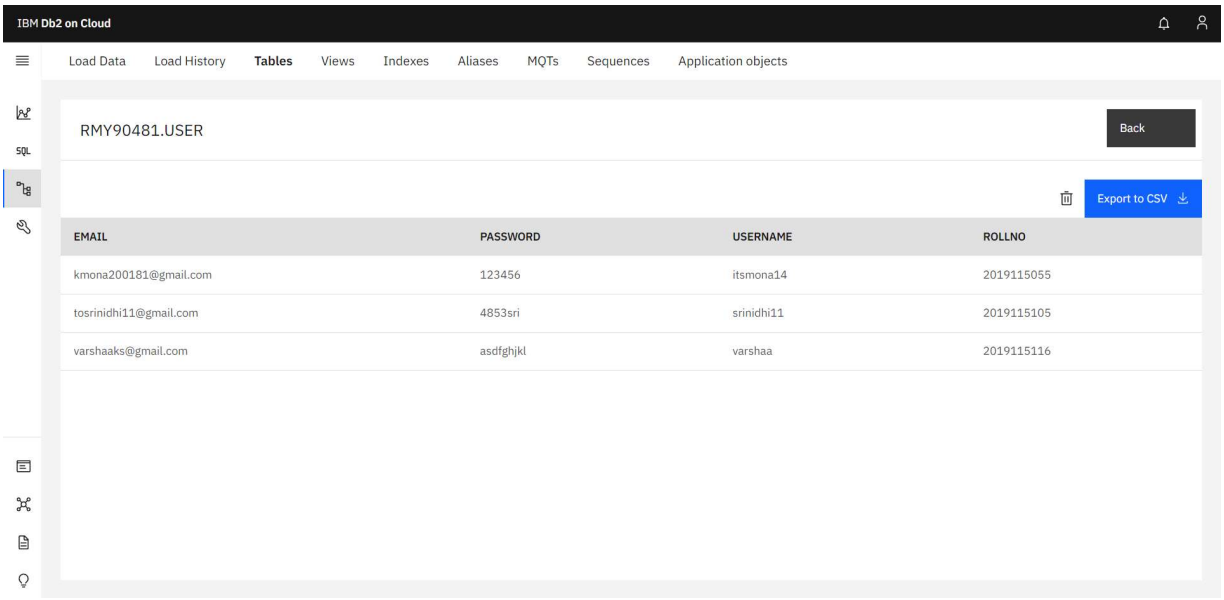
Output:



DELETE

delete from user where rollno = 2019115035



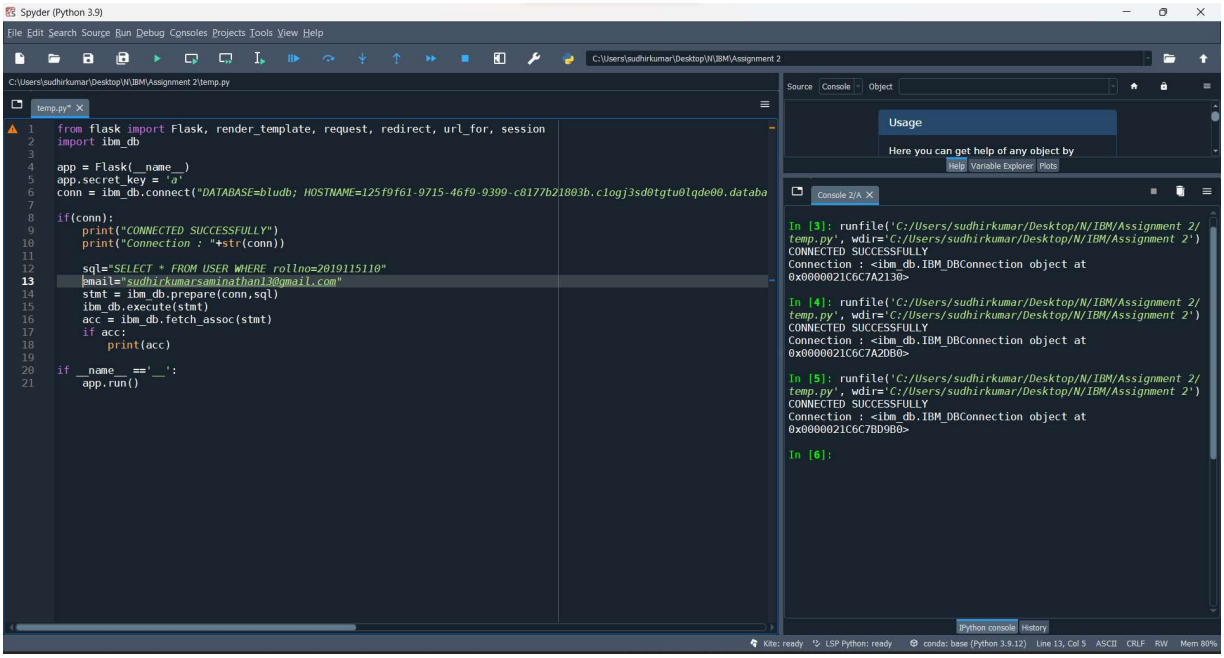


Question 3:

Connect python code to db2.

Solution 3:

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db

app = Flask(__name__)
app.secret_key = 'a'

conn = ibm_db.connect("DATABASE=bludb; HOSTNAME=125f9f61-9715-46f9-9399-
c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud; PORT=30426; SECURITY=SSL;
SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=rmy90481;PWD=qrJOWspW6naGUoOF",'','')

if(conn):
    print("CONNECTED SUCCESSFULLY")
    print("Connection : "+str(conn))

    sql="SELECT * FROM USER WHERE rollno=2019115110"
    email="sudhirkumarsaminathan13@gmail.com"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.execute(stmt)
    acc = ibm_db.fetch_assoc(stmt)
    if acc:
        print(acc)

if __name__ =='__':
    app.run()
```


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\sudhirkumar\Desktop\N\IBM\Assignment 2
C:\Users\sudhirkumar\Desktop\N\IBM\Assignment 2\temp.py
temp.py*
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb; HOSTNAME=125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databa
if(conn):
print("CONNECTED SUCCESSFULLY")
print("Connection : "+str(conn))
sql="SELECT * FROM USER WHERE rollno=2019115110"
email="sudhirkumarsaminathan13@gmail.com"
stmt = ibm_db.prepare(conn,sql)
ibm_db.execute(stmt)
acc = ibm_db.fetch_assoc(stmt)
if acc:
print(acc)
if __name__ =='__':
app.run()
Source Console Object
Usage
Here you can get help of any object by
Help Variable Explorer Plots
Console 2/A
In [3]: runfile('C:/Users/sudhirkumar/Desktop/N/IBM/Assignment 2/temp.py', wdir='C:/Users/sudhirkumar/Desktop/N/IBM/Assignment 2')
CONNECTED SUCCESSFULLY
Connection : <ibm_db.IBM_DBConnection object at 0x0000021C6C7A2130>
In [4]: runfile('C:/Users/sudhirkumar/Desktop/N/IBM/Assignment 2/temp.py', wdir='C:/Users/sudhirkumar/Desktop/N/IBM/Assignment 2')
CONNECTED SUCCESSFULLY
Connection : <ibm_db.IBM_DBConnection object at 0x0000021C6C7A2DB0>
In [5]: runfile('C:/Users/sudhirkumar/Desktop/N/IBM/Assignment 2/temp.py', wdir='C:/Users/sudhirkumar/Desktop/N/IBM/Assignment 2')
CONNECTED SUCCESSFULLY
Connection : <ibm_db.IBM_DBConnection object at 0x0000021C6C7BD9B0>
In [6]:
IPython console History
Kite: ready LSP Python: ready conda: base (Python 3.9.12) Line 13, Col 5 ASCII CRLF RW Mem 80%

Question 4:

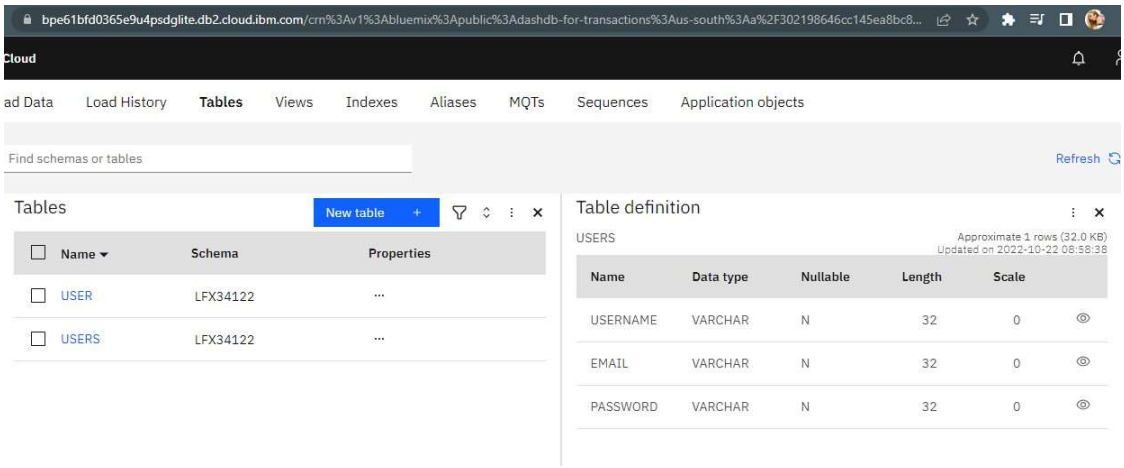
Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid show the welcome page

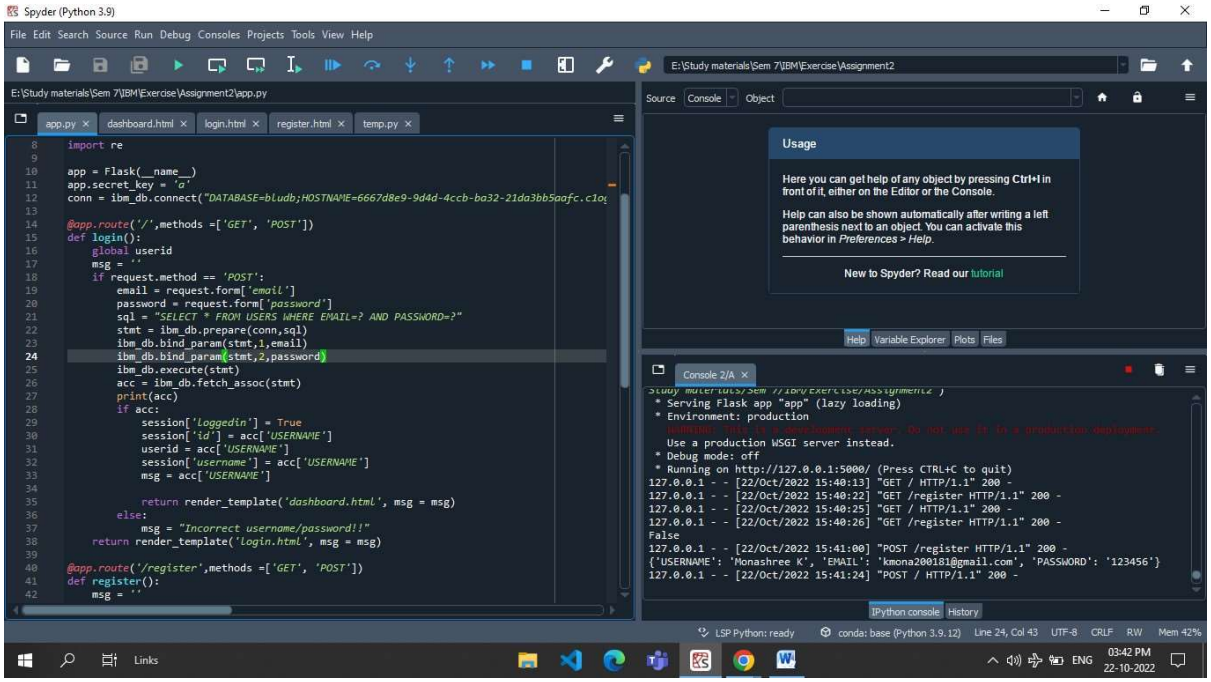
Solution 4:

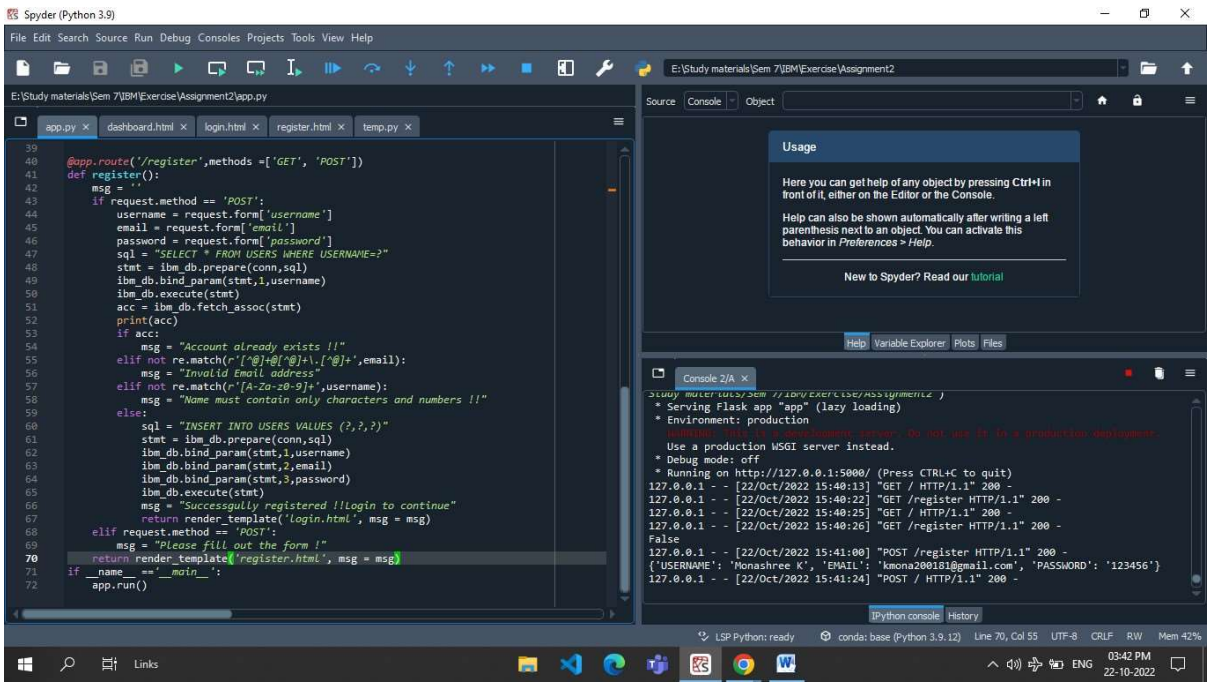
```
# -*- coding: utf-8 -*-

"""

@author: Monashree

"""

from flask import Flask, render_template, request, redirect, url_for, session

import ibm_db

import re

app = Flask(_name_)

app.secret_key = 'a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY=SSL;SSLSe
rverCertificate=DigiCertGlobalRootCA.crt;UID=lfx34122;PWD=jmQDS9wCaxqRIlQd",'','')

@app.route('/',methods =['GET', 'POST'])

def login():

    global userid

    msg = ''

    if request.method == 'POST':

        email = request.form['email']

        password = request.form['password']

        sql = "SELECT * FROM USERS WHERE EMAIL=? AND PASSWORD=?"

        stmt = ibm_db.prepare(conn,sql)

        ibm_db.bind_param(stmt,1,email)

        ibm_db.bind_param(stmt,2,password)

        ibm_db.execute(stmt)
```

```
        acc = ibm_db.fetch_assoc(stmt)
        print(acc)
        if acc:
            session['loggedin'] = True
            session['id'] = acc['USERNAME']
            userid = acc['USERNAME']
            session['username'] = acc['USERNAME']
            msg = acc['USERNAME']

            return render_template('dashboard.html', msg = msg)
        else:
            msg = "Incorrect username/password!!"
    return render_template('login.html', msg = msg)

@app.route('/register',methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM USERS WHERE USERNAME=?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_assoc(stmt)
        print(acc)
```

```
        if acc:
            msg = "Account already exists !!"
        elif not re.match(r'[^@]+@[^@]+\.[^@]+',email):
            msg = "Invalid Email address"
        elif not re.match(r'[A-Za-z0-9]+',username):
            msg = "Name must contain only characters and numbers !!"
        else:
            sql = "INSERT INTO USERS VALUES (?,?,?)"
            stmt = ibm_db.prepare(conn,sql)
            ibm_db.bind_param(stmt,1,username)
            ibm_db.bind_param(stmt,2,email)
            ibm_db.bind_param(stmt,3,password)
            ibm_db.execute(stmt)
            msg = "Successgully registered !!Login to continue"
            return render_template('login.html', msg = msg)
    elif request.method == 'POST':
        msg = "Please fill out the form !"
    return render_template('register.html', msg = msg)
if __name__=='_main_':
    app.run()
```

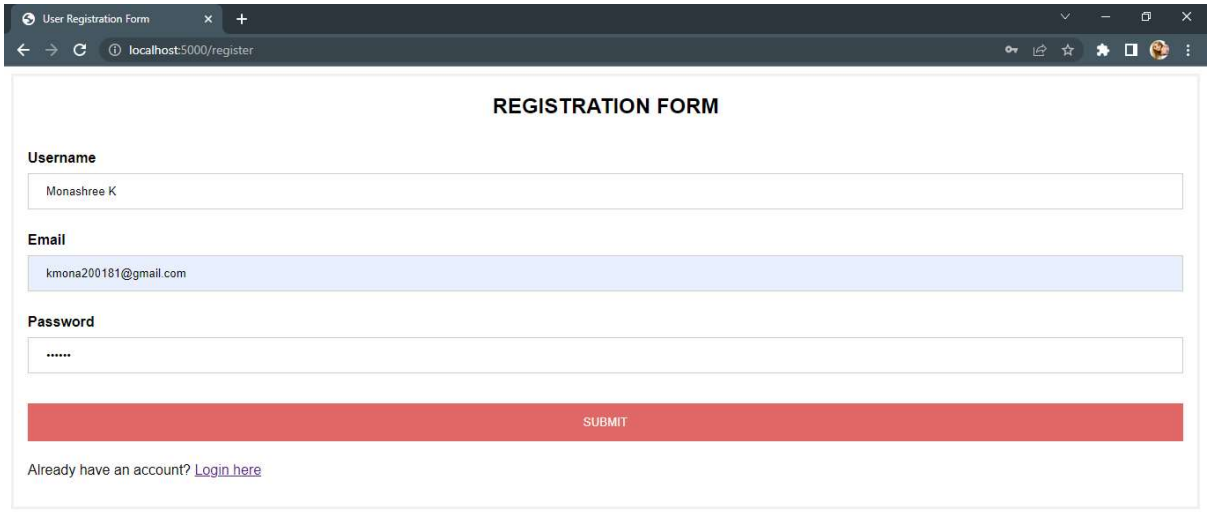



Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
E:\Study materials\Sem 7\IBM\Exercise\Assignment2\app.py
app.py
dashboard.html
login.html
register.html
temp.py
import re
app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1o
@app.route('/',methods =['GET', 'POST'])
def login():
global userid
msg = ''
if request.method == 'POST':
email = request.form['email']
password = request.form['password']
sql = "SELECT * FROM USERS WHERE EMAIL=? AND PASSWORD=?"
stmt = ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,email)
ibm_db.bind_param(stmt,2,password)
ibm_db.execute(stmt)
acc = ibm_db.fetch_assoc(stmt)
print(acc)
if acc:
session['Loggedin'] = True
session['id'] = acc['USERNAME']
userid = acc['USERNAME']
session['username'] = acc['USERNAME']
msg = acc['USERNAME']
return render_template('dashboard.html', msg = msg)
else:
msg = "Incorrect username/password!!"
return render_template('login.html', msg = msg)
@app.route('/register',methods =['GET', 'POST'])
def register():
msg = ''
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help
Variable Explorer
Plots
Files
Console 2/A
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [22/Oct/2022 15:40:13] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [22/Oct/2022 15:40:22] "GET /register HTTP/1.1" 200 -
127.0.0.1 - - [22/Oct/2022 15:40:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [22/Oct/2022 15:40:26] "GET /register HTTP/1.1" 200 -
False
127.0.0.1 - - [22/Oct/2022 15:41:00] "POST /register HTTP/1.1" 200 -
{'USERNAME': 'Monashree K', 'EMAIL': 'kmona200181@gmail.com', 'PASSWORD': '123456'}
127.0.0.1 - - [22/Oct/2022 15:41:24] "POST / HTTP/1.1" 200 -
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 24, Col 43
UTF-8
CRLF
RW
Mem 42%
Links
ENG
03:42 PM
22-10-2022

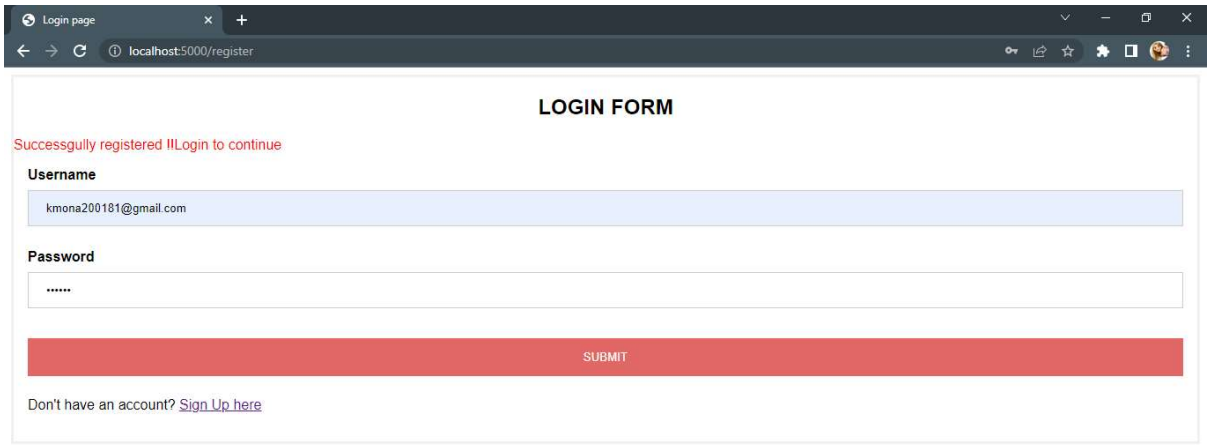

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
E:\Study materials\Sem 7\IBM\Exercise\Assignment2\app.py
app.py
dashboard.html
login.html
register.html
temp.py
@app.route('/register',methods =['GET', 'POST'])
def register():
msg = ''
if request.method == 'POST':
username = request.form['username']
email = request.form['email']
password = request.form['password']
sql = "SELECT * FROM USERS WHERE USERNAME=?"
stmt = ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,username)
ibm_db.execute(stmt)
acc = ibm_db.fetch_assoc(stmt)
print(acc)
if acc:
msg = "Account already exists !!"
elif not re.match(r'[^@]+@[^@]+\.[^@]+',email):
msg = "Invalid Email address"
elif not re.match(r'[A-Za-z0-9]+',username):
msg = "Name must contain only characters and numbers !!"
else:
sql = "INSERT INTO USERS VALUES (?,?,?)"
stmt = ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,username)
ibm_db.bind_param(stmt,2,email)
ibm_db.bind_param(stmt,3,password)
ibm_db.execute(stmt)
msg = "Successqully registered !!Login to continue"
return render_template('login.html', msg = msg)
elif request.method == 'POST':
msg = "Please fill out the form !"
return render_template('register.html', msg = msg)
if __name__ =='__main__':
app.run()
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help
Variable Explorer
Plots
Files
Console 2/A
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [22/Oct/2022 15:40:13] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [22/Oct/2022 15:40:22] "GET /register HTTP/1.1" 200 -
127.0.0.1 - - [22/Oct/2022 15:40:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [22/Oct/2022 15:40:26] "GET /register HTTP/1.1" 200 -
False
127.0.0.1 - - [22/Oct/2022 15:41:00] "POST /register HTTP/1.1" 200 -
{'USERNAME': 'Monashree K', 'EMAIL': 'kmona200181@gmail.com', 'PASSWORD': '123456'}
127.0.0.1 - - [22/Oct/2022 15:41:24] "POST / HTTP/1.1" 200 -
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 70, Col 55
UTF-8
CRLF
RW
Mem 42%
Links
ENG
03:42 PM
22-10-2022

Output:

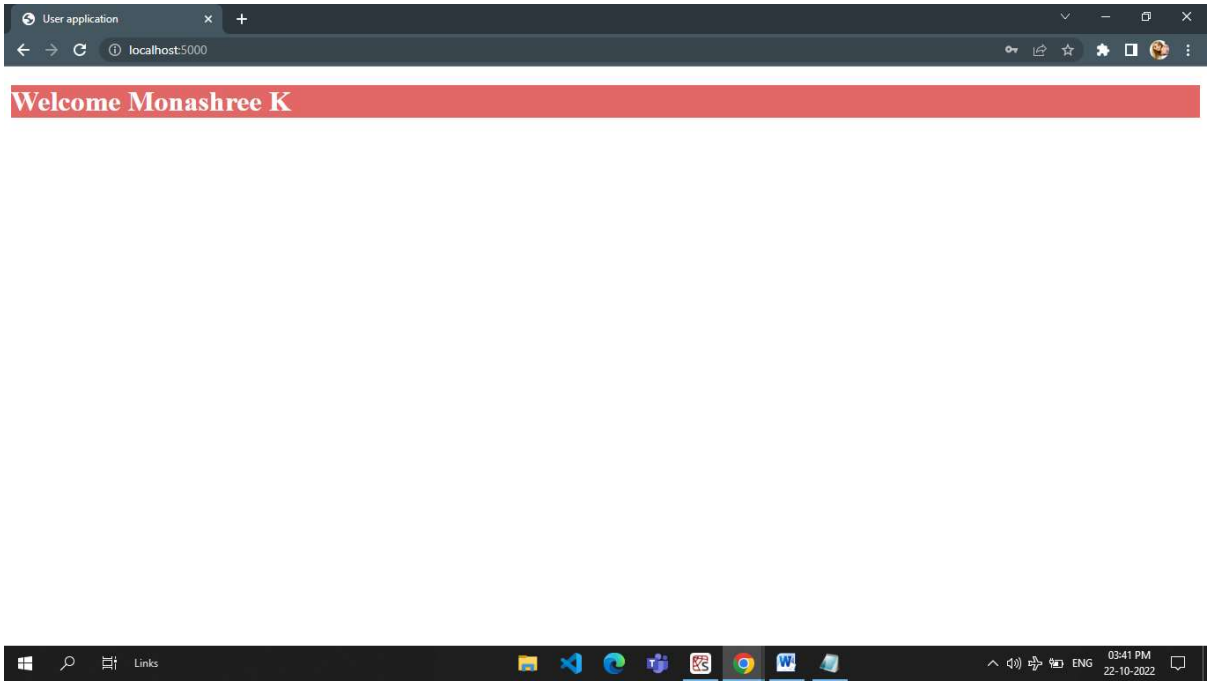
Register page



Login page



Welcome page



Table