Assignment -2

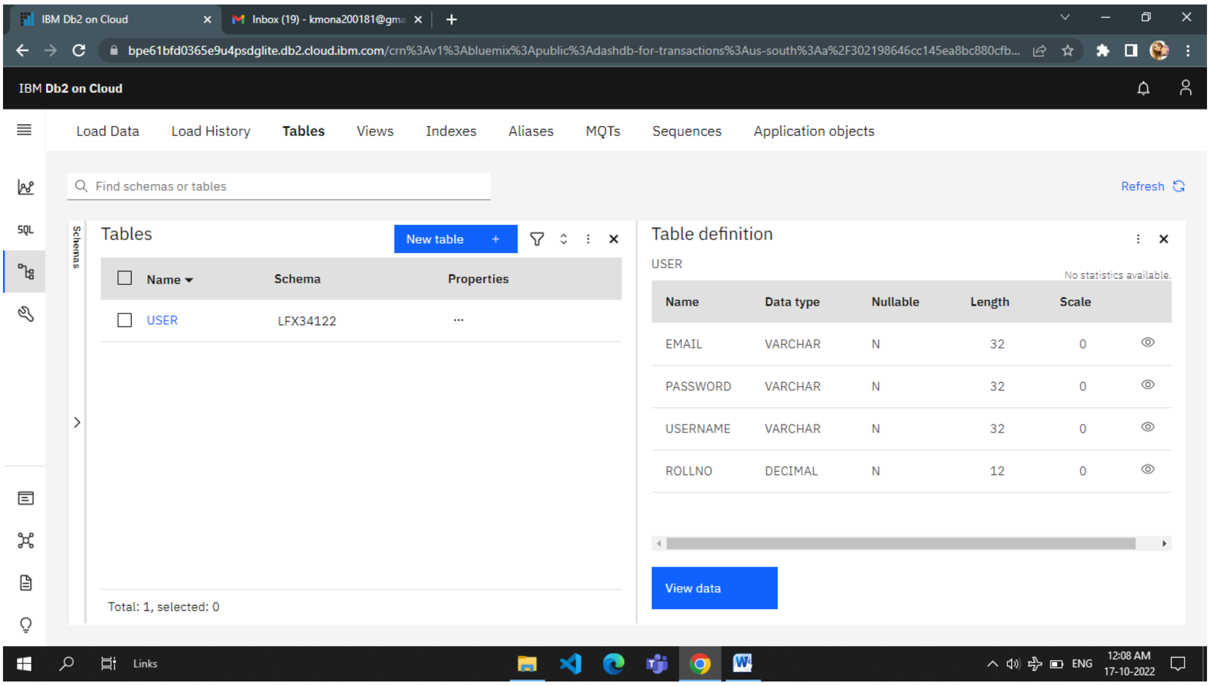
Python Programming

Assignment Date	23 September 2022
Student Name	Monashree K
Student Roll Number	2019115055
Maximum Marks	2 Marks

Question 1:

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

Solution 1:



Tables

Name	Schema	Properties
□ USER	LFX34122	···

Total: 1, selected: 0

Table definition

USER

No statistics available.

Name	Data type	Nullable	Length	Scale
EMAIL	VARCHAR	N	32	0
PASSWORD	VARCHAR	N	32	0
USERNAME	VARCHAR	N	32	0
ROLLNO	DECIMAL	N	12	0

Question 2:

Perform UPDATE, DELETE Queries with user table

Solution 2:

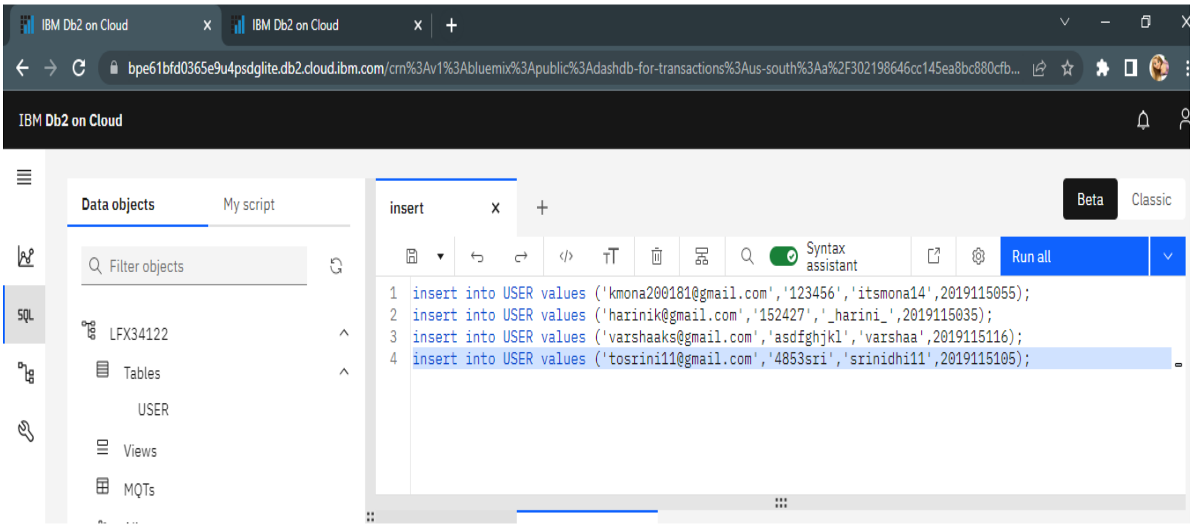
<u>INSERT</u>

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

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

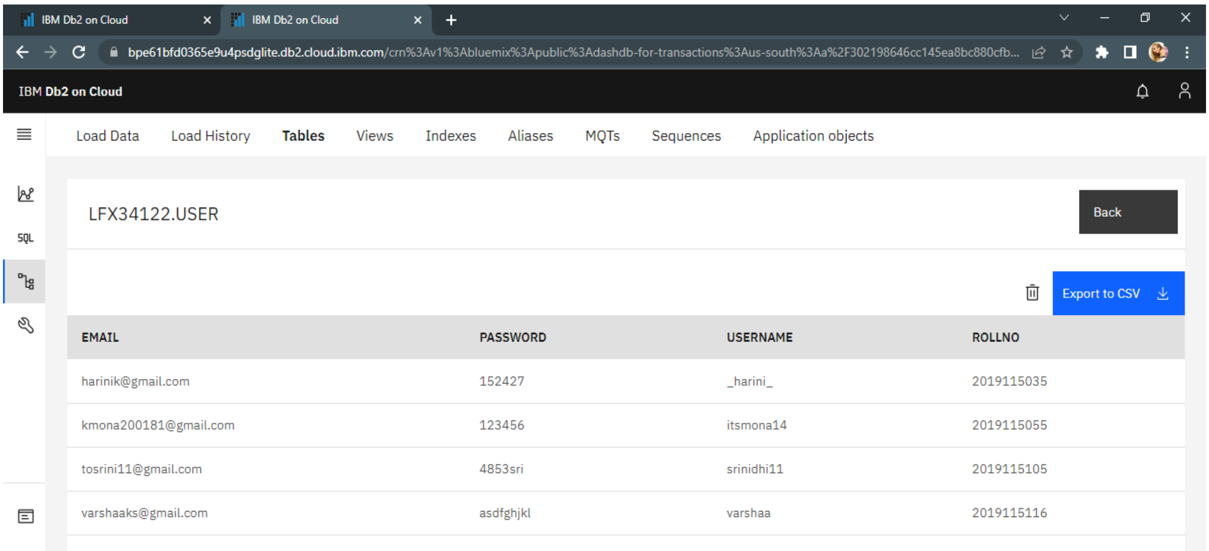
insert into USER values ('varshaaks@gmail.com','asdfghjkl','varshaa',2019115116);

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



```
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:

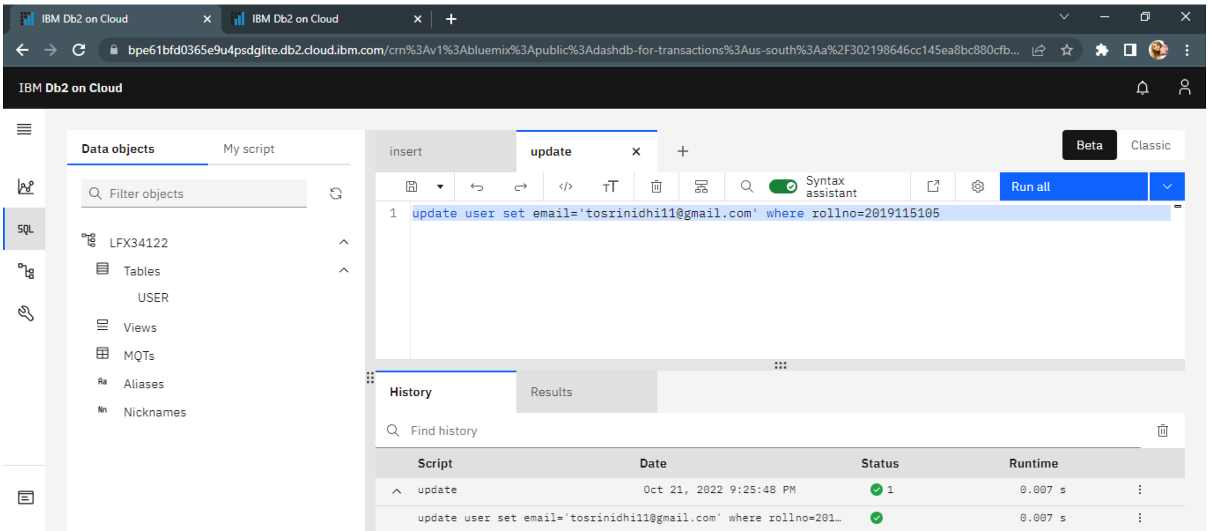


LFX34122.USER

EMAIL	PASSWORD	USERNAME	ROLLNO
harinik@gmail.com	152427	_harini_	2019115035
kmona200181@gmail.com	123456	itsmona14	2019115055
tosrini11@gmail.com	4853sri	srinidhi11	2019115105
varshaaks@gmail.com	asdfghjkl	varshaa	2019115116

UPDATE

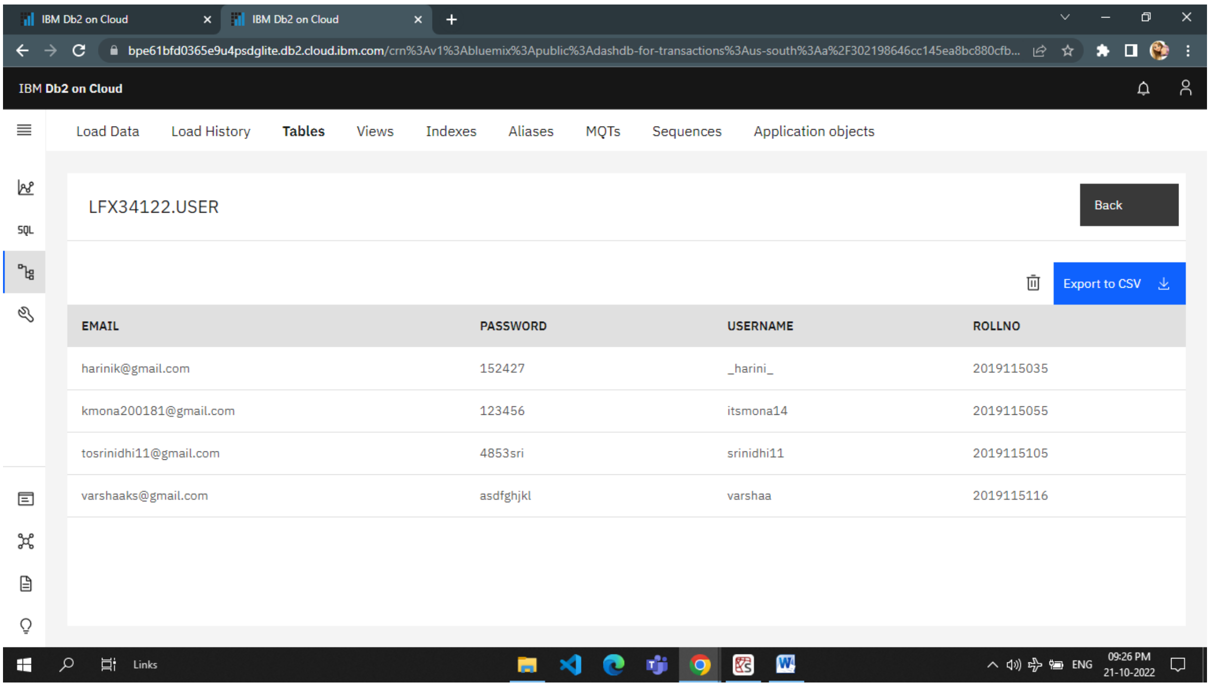
update user set email='tosrinidhi11@gmail.com' where rollno=2019115105



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

Script	Date	Status	Runtime
update	Oct 21, 2022 9:25:48 PM	✓ 1	0.007 s
update user set email='tosrinidhi11@gmail.com' where rollno=201…		✓	0.007 s

Output:

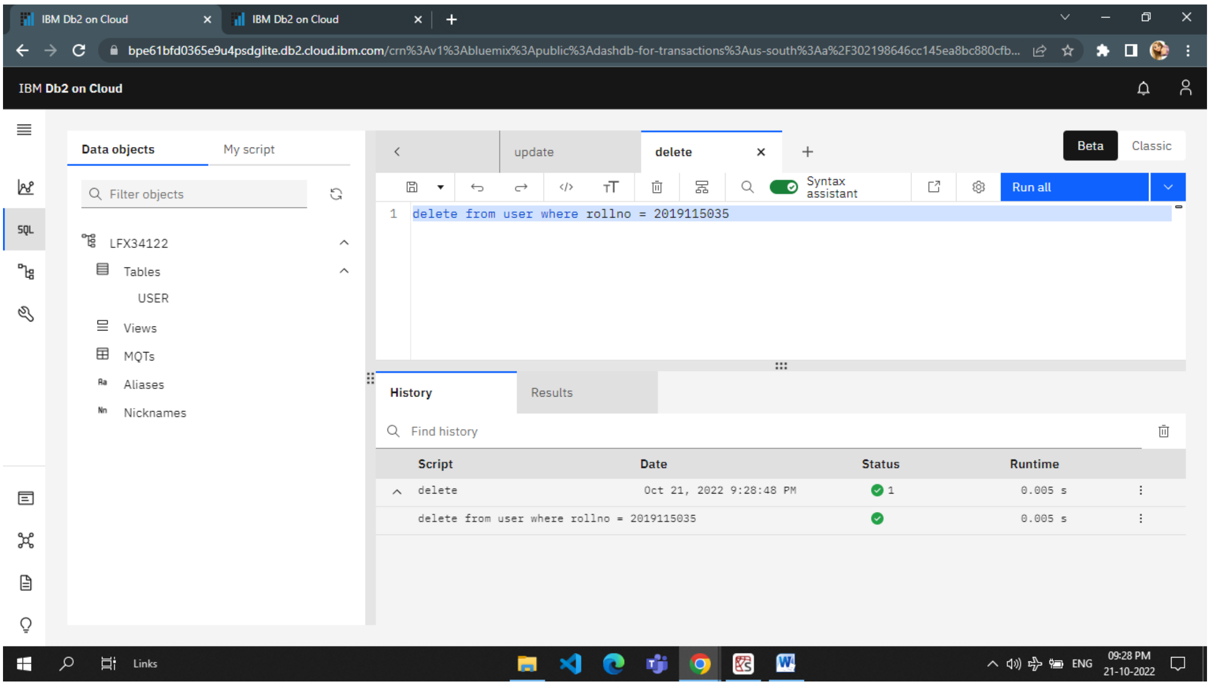


LFX34122.USER

EMAIL	PASSWORD	USERNAME	ROLLNO
harinik@gmail.com	152427	_harini_	2019115035
kmona200181@gmail.com	123456	itsmona14	2019115055
tosrinidhi11@gmail.com	4853sri	srinidhi11	2019115105
varshaaks@gmail.com	asdfghjkl	varshaa	2019115116

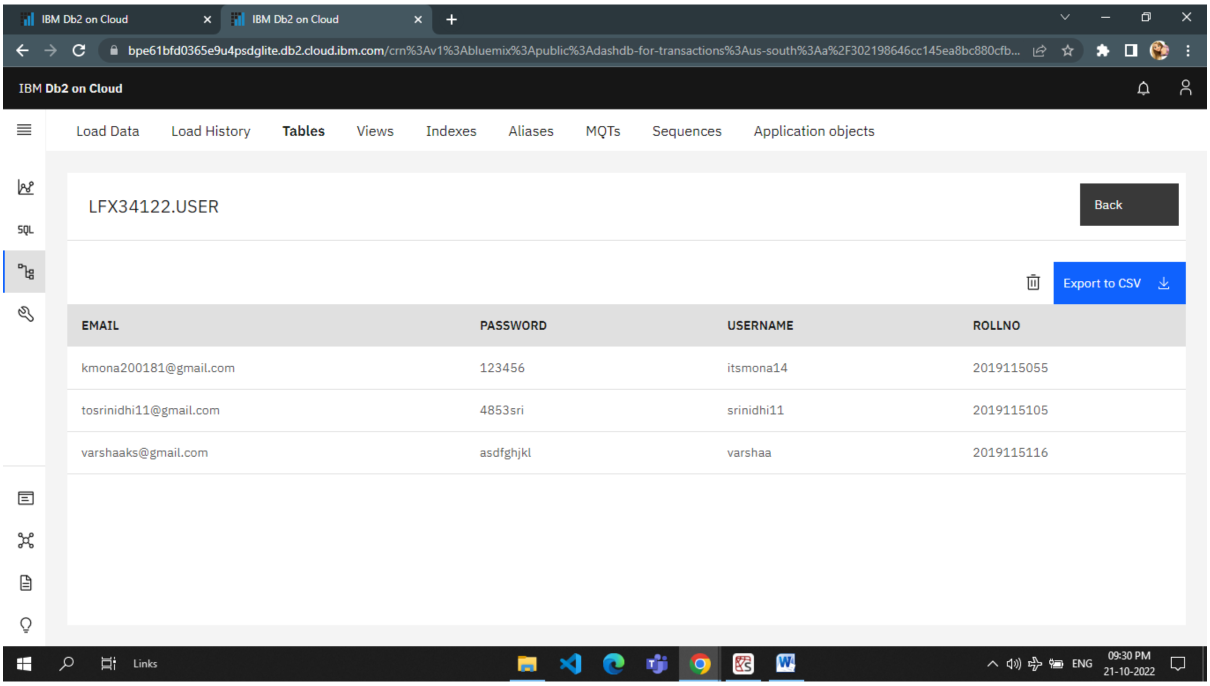
DELETE

delete from user where rollno = 2019115035



```
delete from user where rollno = 2019115035
```

Script	Date	Status	Runtime
delete	Oct 21, 2022 9:28:48 PM	✓ 1	0.005 s
delete from user where rollno = 2019115035		✓	0.005 s



LFX34122.USER

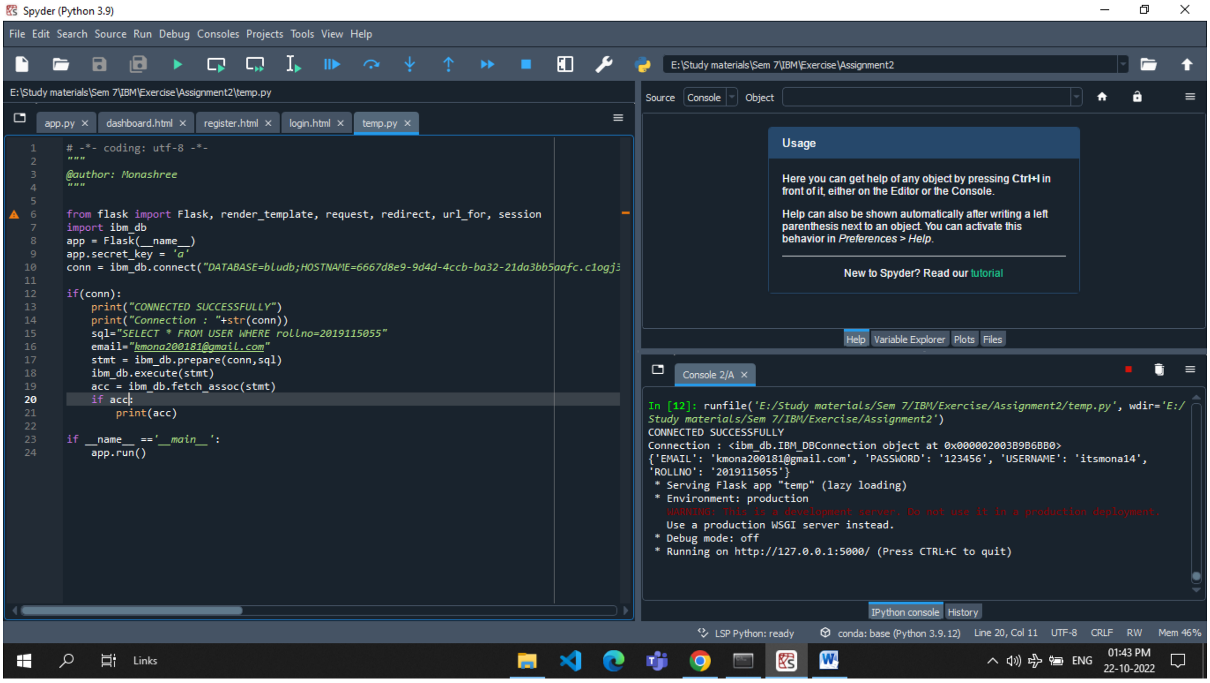
EMAIL	PASSWORD	USERNAME	ROLLNO
kmona200181@gmail.com	123456	itsmona14	2019115055
tosrinidhi11@gmail.com	4853sri	srinidhi11	2019115105
varshaaks@gmail.com	asdfghjkl	varshaa	2019115116

Question 3:

Connect python code to db2.

Solution 3:

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

"""

@author: Monashree

"""

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=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lfx34122;PWD=jmQDS9wCaxqRIlQd",'','')

if(conn):

    print("CONNECTED SUCCESSFULLY")

    print("Connection : "+str(conn))

    sql="SELECT * FROM USER WHERE rollno=2019115055"
    email="kmona200181@gmail.com"

    stmt = ibm_db.prepare(conn,sql)

    ibm_db.execute(stmt)

    acc = ibm_db.fetch_assoc(stmt)

    if acc:

        print(acc)

if __name__ =='__main__':

    app.run()
```



```
# -*- coding: utf-8 -*-
"""
@author: Monashree
"""

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=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3

if(conn):
    print("CONNECTED SUCCESSFULLY")
    print("Connection : "+str(conn))
    sql="SELECT * FROM USER WHERE rollno=2019115055"
    email="kmona200181@gmail.com"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.execute(stmt)
    acc = ibm_db.fetch_assoc(stmt)
    if acc:
        print(acc)

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

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

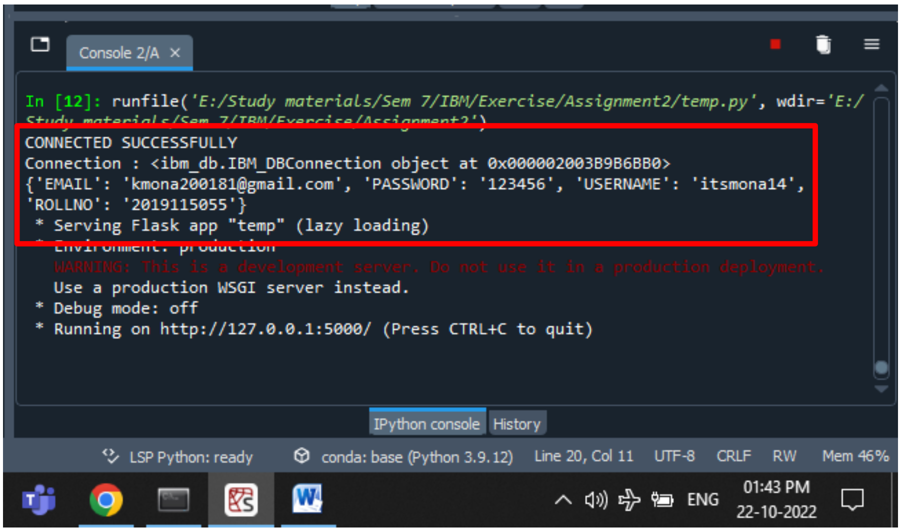
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
In [12]: runfile('E:/Study materials/Sem 7/IBM/Exercise/Assignment2/temp.py', wdir='E:/
Study materials/Sem 7/IBM/Exercise/Assignment2')
CONNECTED SUCCESSFULLY
Connection : <ibm_db.IBM_DBConnection object at 0x000002003B9B6BB0>
{'EMAIL': 'kmona200181@gmail.com', 'PASSWORD': '123456', 'USERNAME': 'itsmona14',
'ROLLNO': '2019115055'}
 * Serving Flask app "temp" (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)
```

LSP Python: ready   conda: base (Python 3.9.12)   Line 20, Col 11   UTF-8   CRLF   RW   Mem 46%

Output:



```
In [12]: runfile('E:/Study materials/Sem 7/IBM/Exercise/Assignment2/temp.py', wdir='E:/
Study materials/Sem 7/IBM/Exercise/Assignment2')
CONNECTED SUCCESSFULLY
Connection : <ibm_db.IBM_DBConnection object at 0x000002003B9B6BB0>
{'EMAIL': 'kmona200181@gmail.com', 'PASSWORD': '123456', 'USERNAME': 'itsmona14',
'ROLLNO': '2019115055'}
 * Serving Flask app "temp" (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)
```

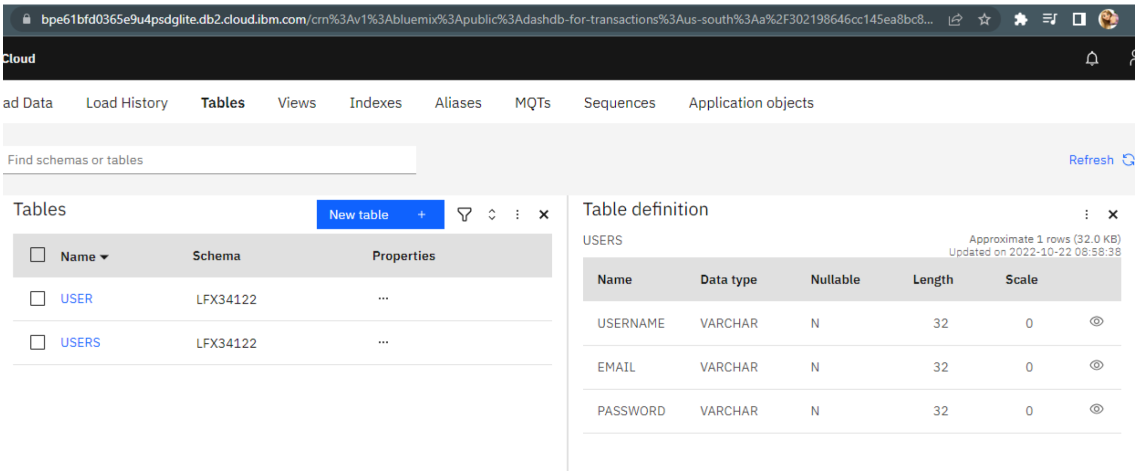
LSP Python: ready   conda: base (Python 3.9.12)   Line 20, Col 11   UTF-8   CRLF   RW   Mem 46%

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;SSLServerCertificate=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()
```



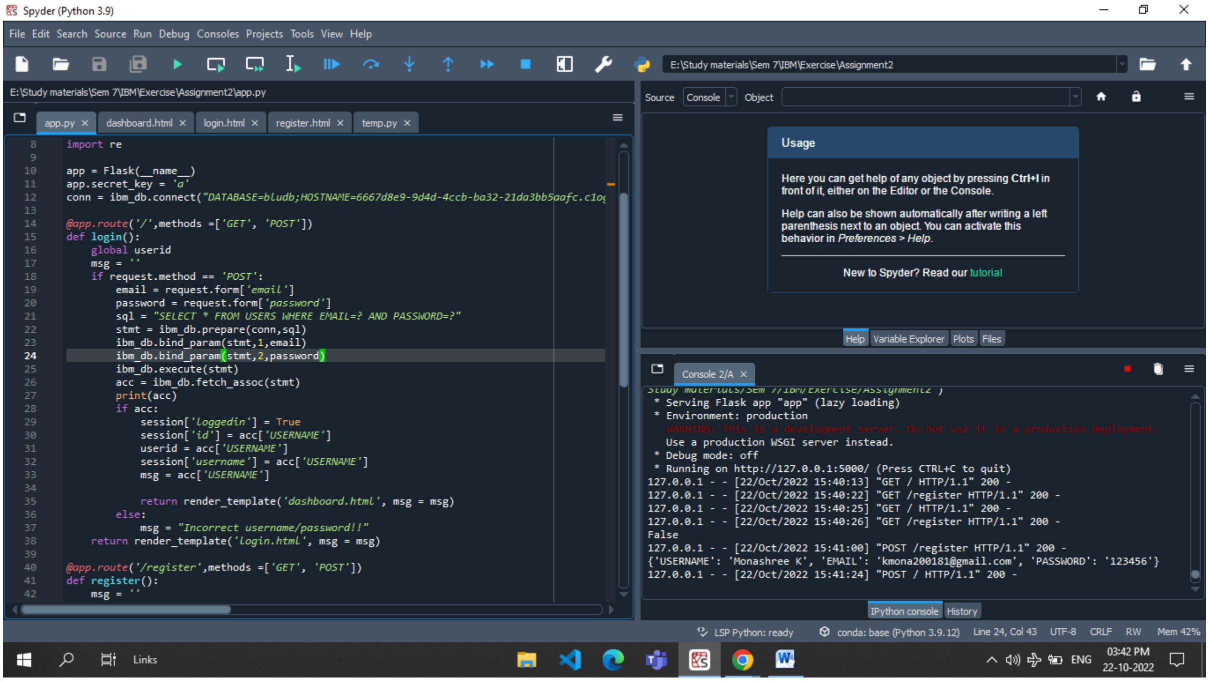
bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F302198646cc145ea8bc8...

☐	Name ▾	Schema	Properties
☐	USER	LFX34122	···
☐	USERS	LFX34122	···

USERS

Approximate 1 rows (32.0 KB)
Updated on 2022-10-22 08:58:38

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	N	32	0
EMAIL	VARCHAR	N	32	0
PASSWORD	VARCHAR	N	32	0



Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
E:\Study materials\Sem 7\IBM\Exercise\Assignment2
E:\Study materials\Sem 7\IBM\Exercise\Assignment2\app.py
app.py dashboard.html login.html register.html temp.py
Source Console Object
Help Variable Explorer Plots Files
Console 2/A
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

Usage

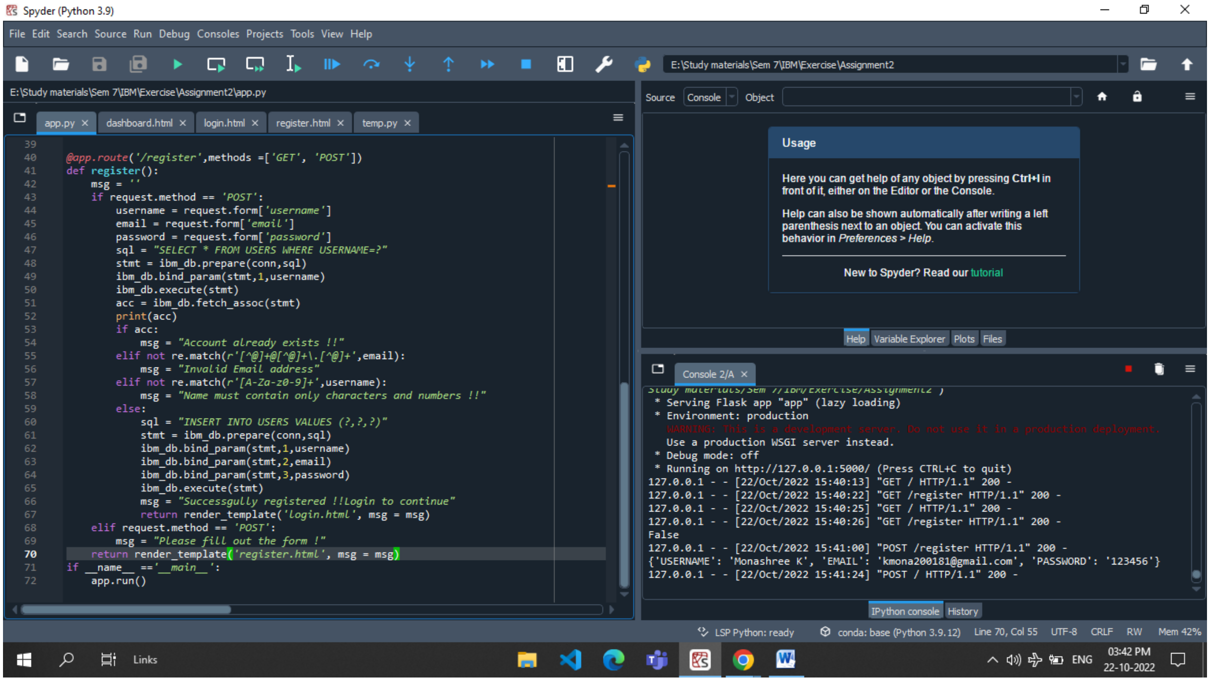
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

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

```python
import re

app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1og

@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 = ''
```

```
 * 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 -
```



Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
E:\Study materials\Sem 7\IBM\Exercise\Assignment2
E:\Study materials\Sem 7\IBM\Exercise\Assignment2\app.py
app.py dashboard.html login.html register.html temp.py
Source Console Object
Help Variable Explorer Plots Files
Console 2/A
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

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*.

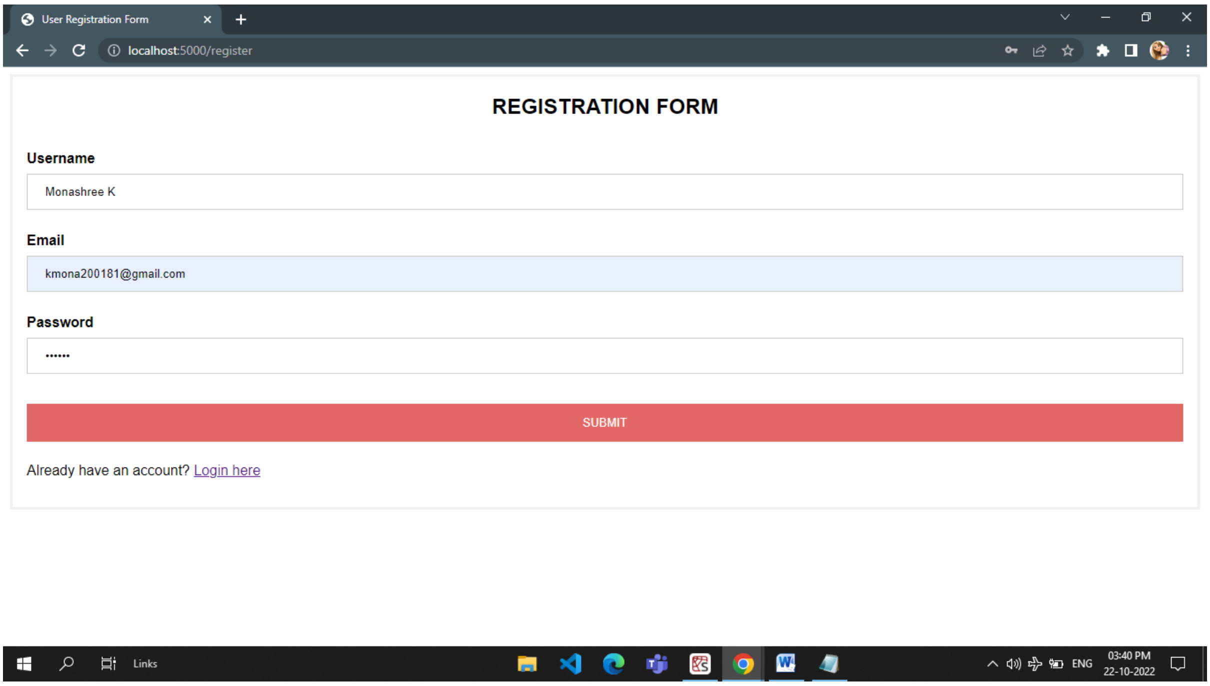
New to Spyder? Read our tutorial

```python

@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()
```

```
 * 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 -
```

Output:

Register page



REGISTRATION FORM

Username

Monashree K

Email

kmona200181@gmail.com

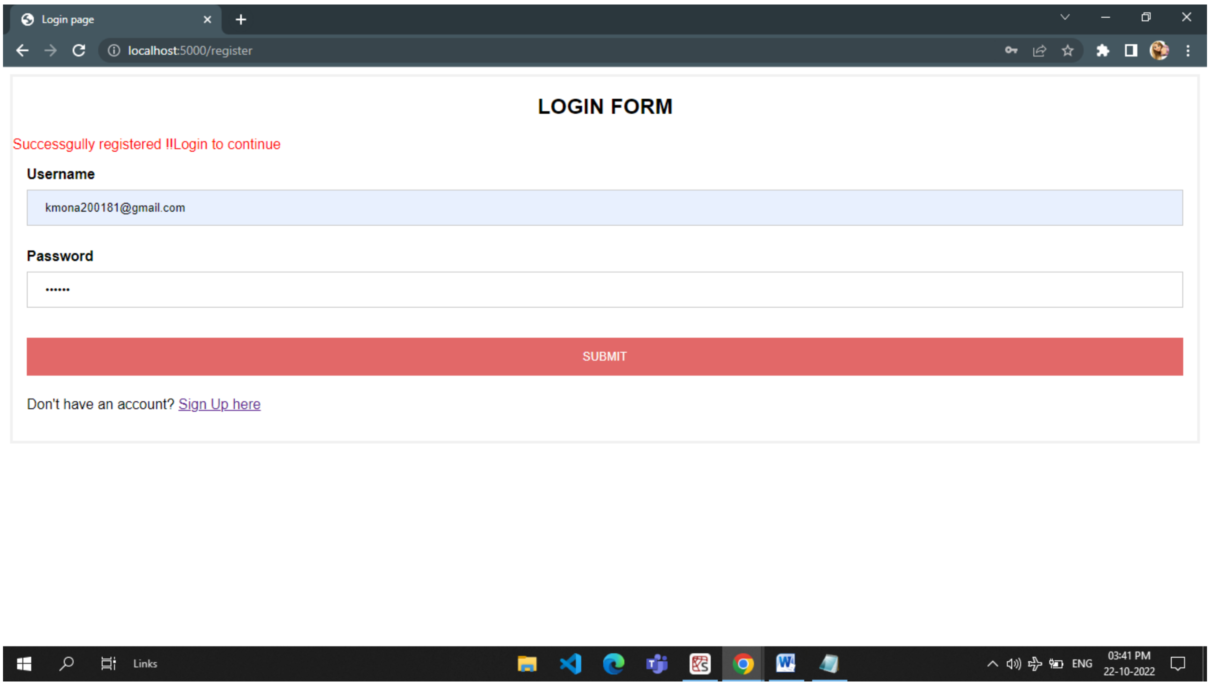
Password

••••••

SUBMIT

Already have an account? Login here

Login page



LOGIN FORM

Successgully registered !!Login to continue

Username

kmona200181@gmail.com

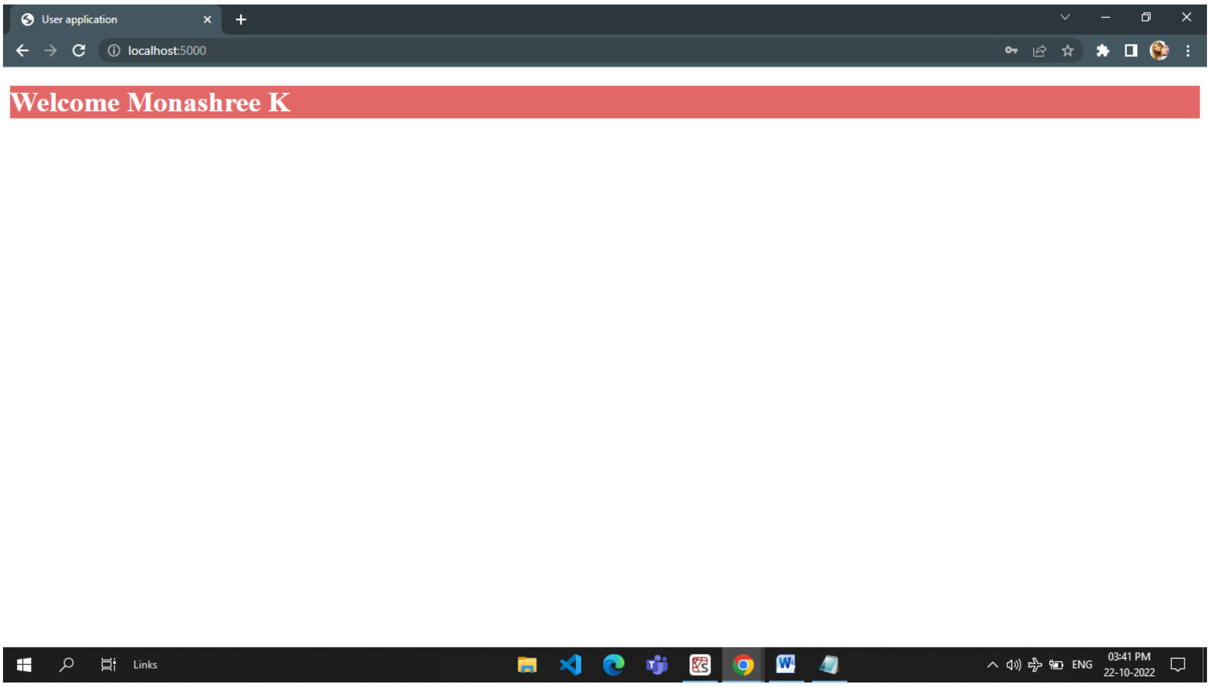
Password

••••••

SUBMIT

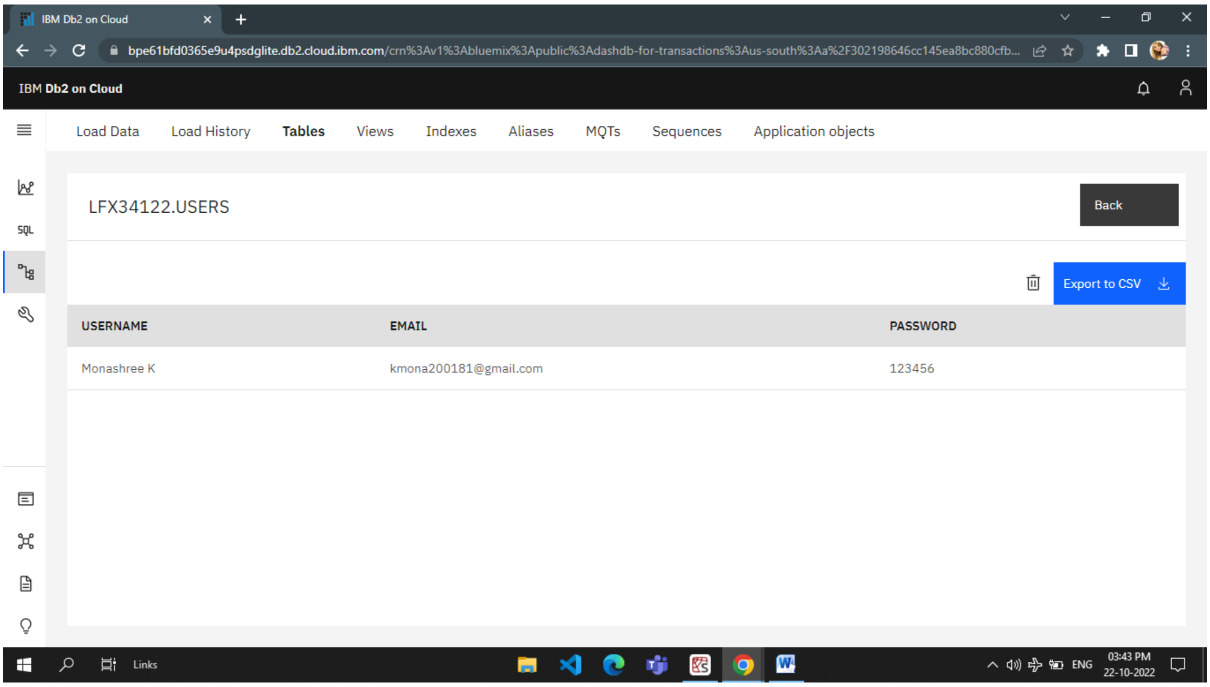
Don't have an account? Sign Up here

Welcome page



Welcome Monashree K

Table



LFX34122.USERS

USERNAME	EMAIL	PASSWORD
Monashree K	kmona200181@gmail.com	123456