

IBM DB2 with PYTHON

Team ID	PNT2022TMID41314
Project Name	Smart Fashion Recommender Application

TASK:

Create IBM DB2 and Connect with Python

```
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=b70a  
f05b-76e4-4bca-a1f5-  
23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud  
;PORT=32716;SECURITY=SSL;SSLServerCertificate=DigiCert
```

```
GlobalRootCA.crt;UID=kmz12090;PWD=Y3bxFYobLHdUjeXy  
;","")
```

```
@app.route("/", 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)
```

```
        account = ibm_db.fetch_assoc(stmt)
```

```
        print(account)
```

```
        if account:
```

```
            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:
        insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, username)
        ibm_db.bind_param(prepare_stmt, 2, email)
        ibm_db.bind_param(prepare_stmt, 3, password)
        ibm_db.execute(prepare_stmt)

        msg = 'You have successfully registered !'

    elif request.method == 'POST':
        msg = 'Please fill out the form !'

    return render_template('register.html', msg=msg)

```

```

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

```

```

def login():

```

```

    global userid

```

```

    msg = "

```

```

    if request.method == 'POST':

```

```

        username = request.form['username']

```

```
password = request.form['password']

sql = "SELECT * FROM users WHERE username =? AND
password=?"

stmt = ibm_db.prepare(conn, sql)

ibm_db.bind_param(stmt, 1, username)

ibm_db.bind_param(stmt, 2, password)

ibm_db.execute(stmt)

account = ibm_db.fetch_assoc(stmt)

print(account)

if account:

    session['loggedin'] = True

    session['id'] = account['USERNAME']

    userid = account['USERNAME']

    session['username'] = account['USERNAME']

    msg = 'Logged in successfully !'

    msg = 'Logged in successfully !'

    return render_template('dashboard.html', msg=msg)

else:

    msg = 'Incorrect username / password !'

return render_template('login.html', msg=msg)
```

```
if __name__ == '__main__':  
    app.run(host='0.0.0.0')  
    # app.run(debug=True)
```



```

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=b70af05b-76e4-4bca-a1f5-
23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;SECURITY=SSL;SS
LServerCertificate=DigiCertGlobalRootCA.crt;UID=kmz12090;PWD=Y3bxFYobLHdUjeXy;", '', '')

@app.route("/", 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)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            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:
            insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, username)
            ibm_db.bind_param(prepare_stmt, 2, email)
            ibm_db.bind_param(prepare_stmt, 3, password)
            ibm_db.execute(prepare_stmt)
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg=msg)

@app.route('/login', methods=['GET', 'POST'])
def login():
    global userid
    msg = ''

    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE username =? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)

```

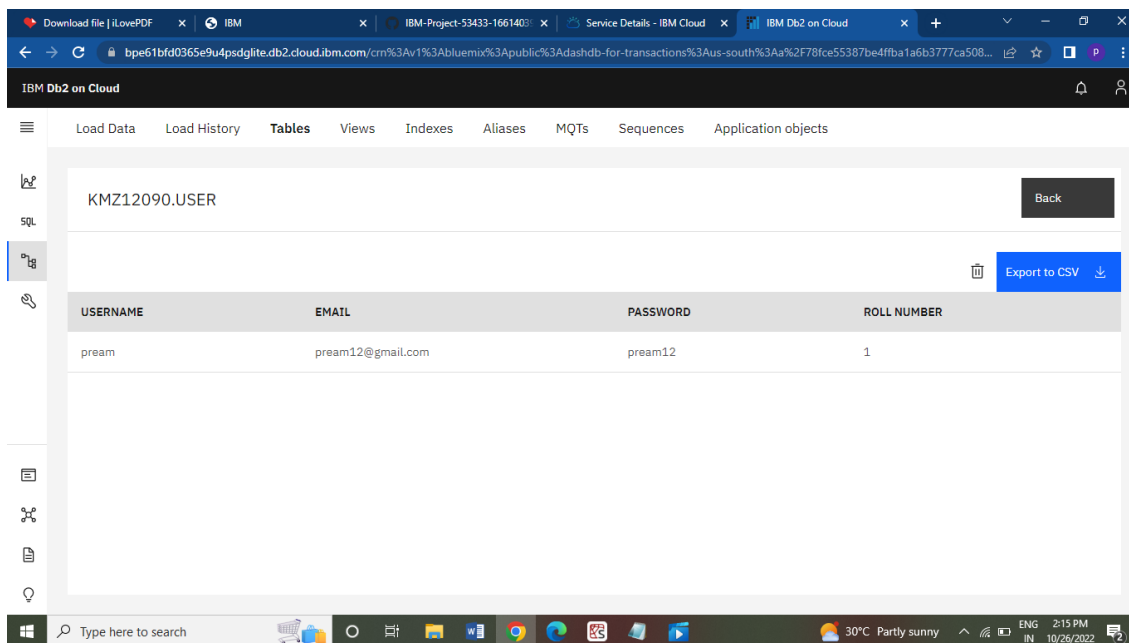
```

if account:
    session['loggedin'] = True
    session['id'] = account['USERNAME']
    userid = account['USERNAME']
    session['username'] = account['USERNAME']
    msg = 'Logged in successfully !'

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

if __name__ == '__main__':
    app.run(host='0.0.0.0')
    # app.run(debug=True)

```



The screenshot shows the IBM Db2 on Cloud web interface. The browser address bar displays a URL starting with 'bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com'. The interface has a top navigation bar with tabs: Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The 'Tables' tab is selected. On the left, there is a sidebar with icons for SQL, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The main content area shows a table with the title 'KMZ12090.USER'. The table has four columns: USERNAME, EMAIL, PASSWORD, and ROLL NUMBER. The first row of data shows 'pream', 'pream12@gmail.com', 'pream12', and '1'. There are 'Back', 'Export to CSV', and 'Download' buttons in the top right corner of the table area. The Windows taskbar at the bottom shows the date as 10/26/2022 and the time as 2:15 PM.

USERNAME	EMAIL	PASSWORD	ROLL NUMBER
pream	pream12@gmail.com	pream12	1