

## ASSIGNMENT 3

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

```
mysql> select * from user;
+-----+-----+-----+-----+
| username | email          | rollno | password |
+-----+-----+-----+-----+
| AB       | ab@gmail.com   | 4      | 01       |
| XY       | xy@gmail.com   | 4      | 03       |
| UV       | uv@gmail.com   | 4      | 04       |
| UV       | uv@gmail.com   | 3      | 04       |
| AB       | ab@gmail.com   | 2      | 03       |
| CD       | cd@gmail.com   | 8      | 05       |
| MN       | mn@gmail.com   | 0      | 09       |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> delete from user where rollno=4;
Query OK, 3 rows affected (0.00 sec)

mysql> select * from user;
+-----+-----+-----+-----+
| username | email          | rollno | password |
+-----+-----+-----+-----+
| UV       | uv@gmail.com   | 3      | 04       |
| AB       | ab@gmail.com   | 2      | 03       |
| CD       | cd@gmail.com   | 8      | 05       |
| MN       | mn@gmail.com   | 0      | 09       |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> update user
-> set password=00
-> where username='cd';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from user;
+-----+-----+-----+-----+
| username | email          | rollno | password |
+-----+-----+-----+-----+
| UV       | uv@gmail.com   | 3      | 04       |
| AB       | ab@gmail.com   | 2      | 03       |
| CD       | cd@gmail.com   | 8      | 0       |
| MN       | mn@gmail.com   | 0      | 09       |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

CREATE TABLE:

CREATE TABLE Student (Username varchar (20), email varchar (20), roll number int(15),password varchar(30));

INSERT QUERY:

Insert into Student values ('uv', 'uv@gmail.com','3','04');

Insert into Student values ('AB', 'AB@gmail.com','2','03');

Insert into Student values ('CD', 'CD@gmail.com','8','05');

Insert into Student values ('MN', 'MN@gmail.com','0','09');

## 2.perform UPDATE DELETE queries with user table?

Command Prompt - mysql -u root -p

username	email	rollno	password
AB	ab@gmail.com	4	01
XY	xy@gmail.com	4	03
UV	uv@gmail.com	4	04
UV	uv@gmail.com	3	04
AB	ab@gmail.com	2	03
CD	cd@gmail.com	8	05
MN	mn@gmail.com	0	09

7 rows in set (0.00 sec)

```
mysql> delete from user where rollno='4';  
Query OK, 3 rows affected (0.00 sec)
```

```
mysql> select * from user;
```

username	email	rollno	password
UV	uv@gmail.com	3	04
AB	ab@gmail.com	2	03
CD	cd@gmail.com	8	05
MN	mn@gmail.com	0	09

4 rows in set (0.00 sec)

```
mysql> update user
```

```
    -> set password=00
```

```
    -> where username='cd';
```

```
Query OK, 1 row affected (0.00 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

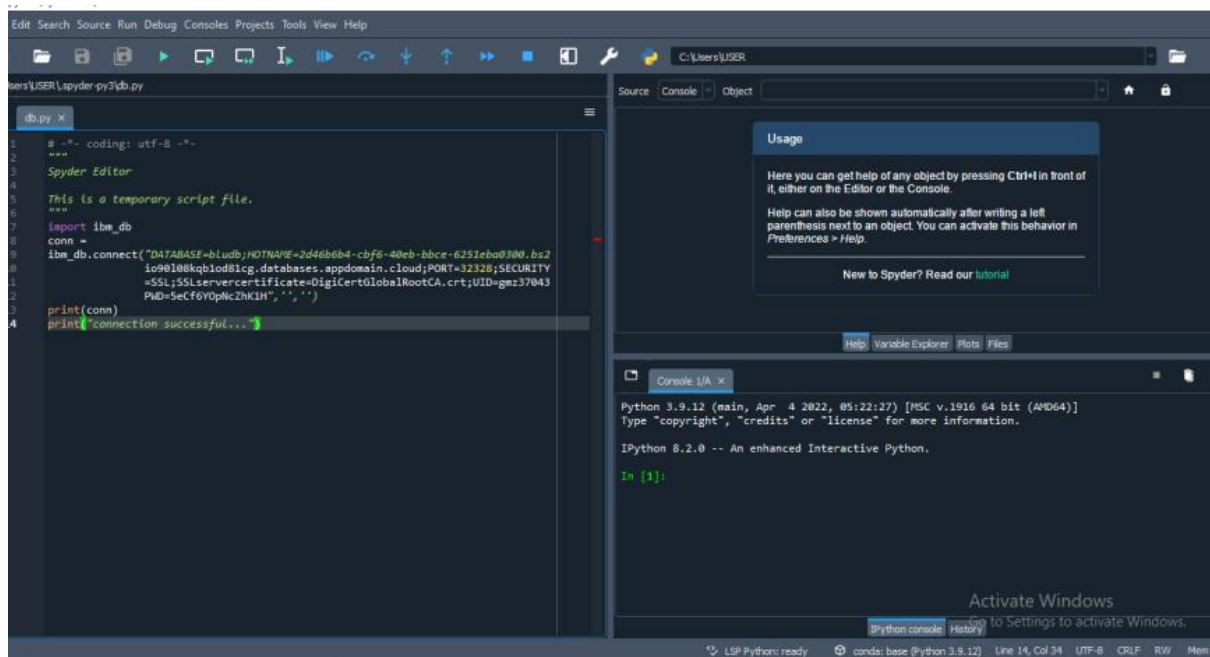
```
mysql> select * from user;
```

username	email	rollno	password
UV	uv@gmail.com	3	04
AB	ab@gmail.com	2	03
CD	cd@gmail.com	8	0
MN	mn@gmail.com	0	09

4 rows in set (0.00 sec)

```
mysql>
```

### 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

The screenshot shows the Spyder Python IDE with a Flask application. The code in the editor is as follows:

```

1 # -- coding: utf-8 --
2
3 Spyder Editor
4
5 This is a temporary script file.
6
7
8 from flask import Flask, render_template, request, redirect, url_for, session
9 import ibm_db
10 import re
11
12 app = Flask(name)
13
14 app.secret_key = 'a'
15
16 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b70af05b-76e4-4bca-a1f5-23dbb4c6a7
17
18 @app.route('/')
19
20 def home():
21     return render_template('home.html')
22
23
24 @app.route('/login', methods=['GET', 'POST'])
25 def login():
26     global userid
27     msg = ''
28     if request.method == 'POST':
29         username = request.form['username']
30         password = request.form['password']
31         sql = "SELECT * FROM users WHERE username =? AND password=?"
32         stmt = ibm_db.prepare(conn, sql)
33         ibm_db.bind_param(stmt, 1, username)
34         ibm_db.bind_param(stmt, 2, password)
35         ibm_db.execute(stmt)

```

The right-hand side of the IDE shows the 'Usage' panel with help text and the 'Console' panel with the following output:

```

Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]:

```

The screenshot shows the Spyder Python IDE with a Flask application. The code in the editor is as follows:

```

36 account = ibm_db.fetch_assoc(stmt)
37 print(account)
38 if account:
39     session['loggedin'] = True
40     session['id'] = account['USERNAME']
41     userid = account['USERNAME']
42     session['username'] = account['USERNAME']
43     msg = 'logged in successfully !'
44
45     msg = 'Logged in successfully !'
46     return render_template('dashboard.html', ..
47
48 mam ipo connect paniducha da
49 db2 ha python kooda
50 @app.route('/register', methods=['GET', 'POST'])
51 def register():
52     msg = ''
53     if request.method == 'POST':
54         username = request.form['username']
55         email = request.form['email']
56         password = request.form['password']
57         sql = "SELECT * FROM users WHERE username =?"
58         stmt = ibm_db.prepare(conn, sql)
59         ibm_db.bind_param(stmt, 1, username)
60         ibm_db.execute(stmt)
61         account = ibm_db.fetch_assoc(stmt)
62         print(account)
63
64 Pannitu irukaga
65 if account:
66     msg = 'Account already exists !'
67
68 elif not re.match(r'([a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z0-9]+)', email):
69     msg = 'Invalid email address !'
70
71 elif not re.match(r'[a-zA-Z0-9]+', username):
72     msg = 'name must contain only characters and numbers !'
73
74 else:
75     insert_sql = "INSERT INTO users VALUES (?, ?, ?)"

```

The right-hand side of the IDE shows the 'Usage' panel with help text and the 'Console' panel with the following output:

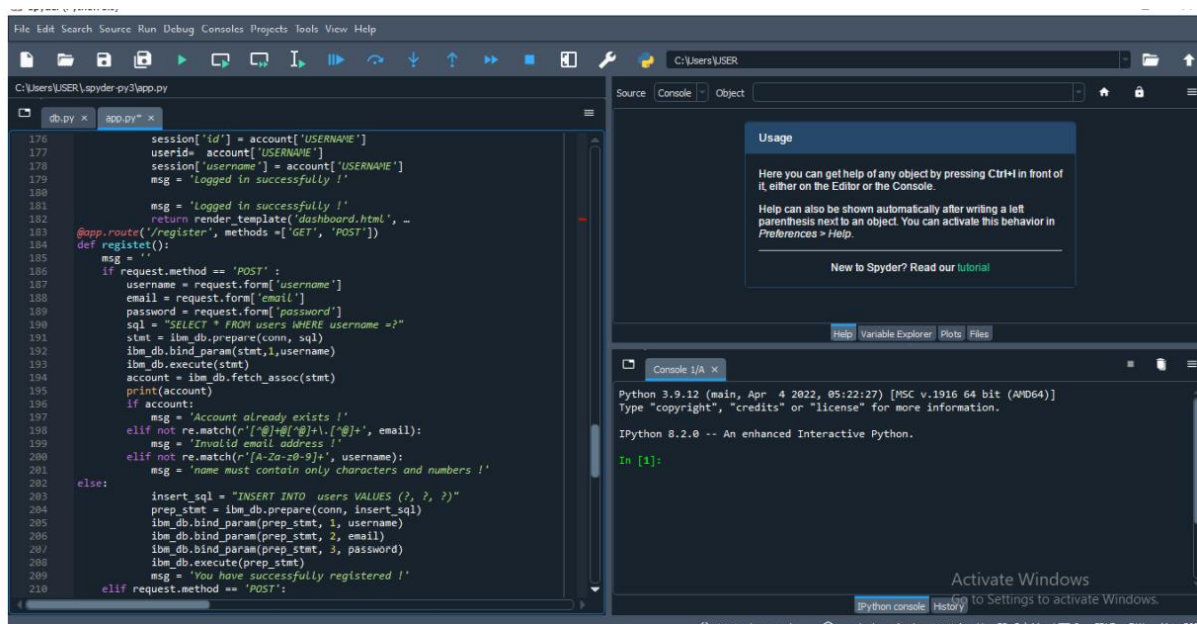
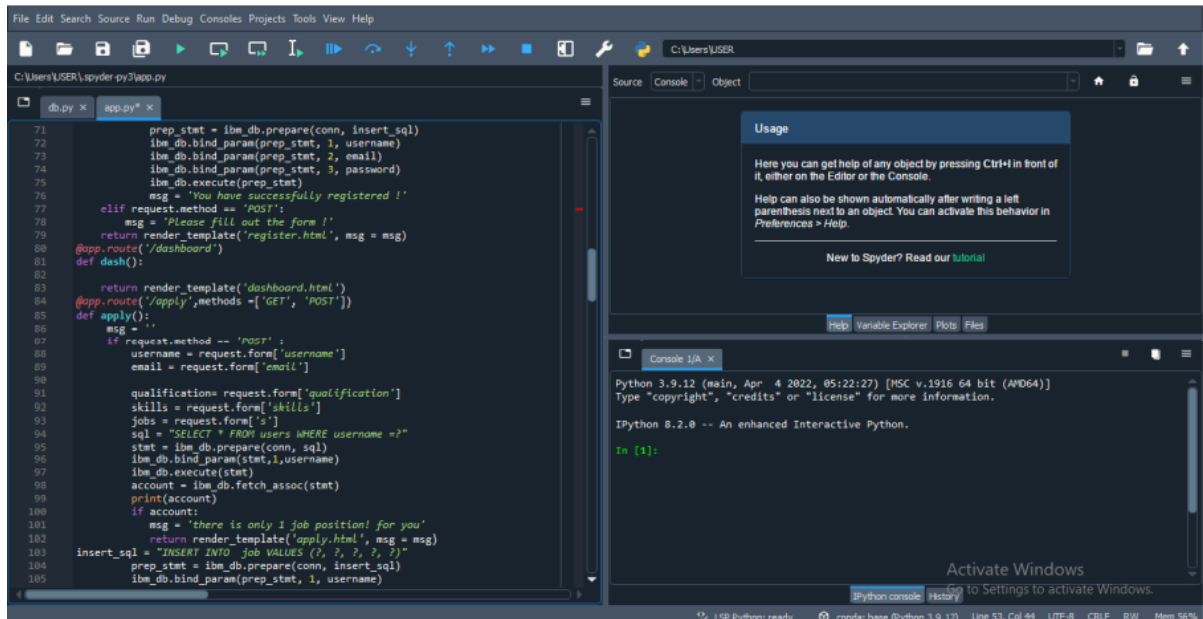
```

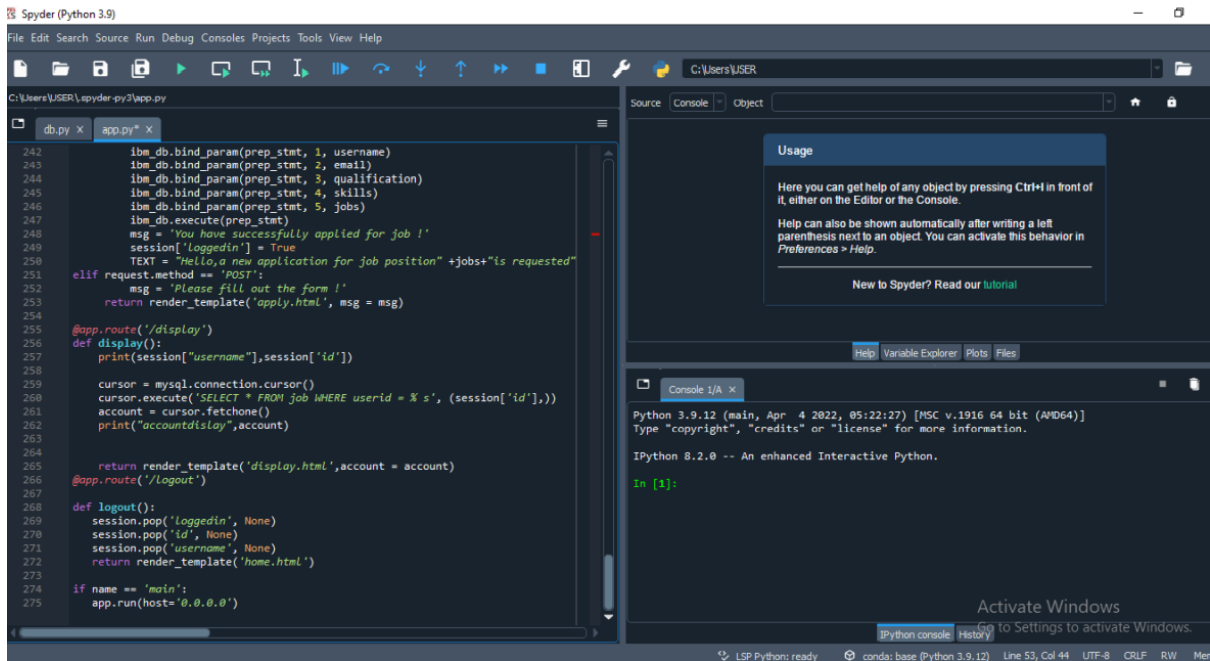
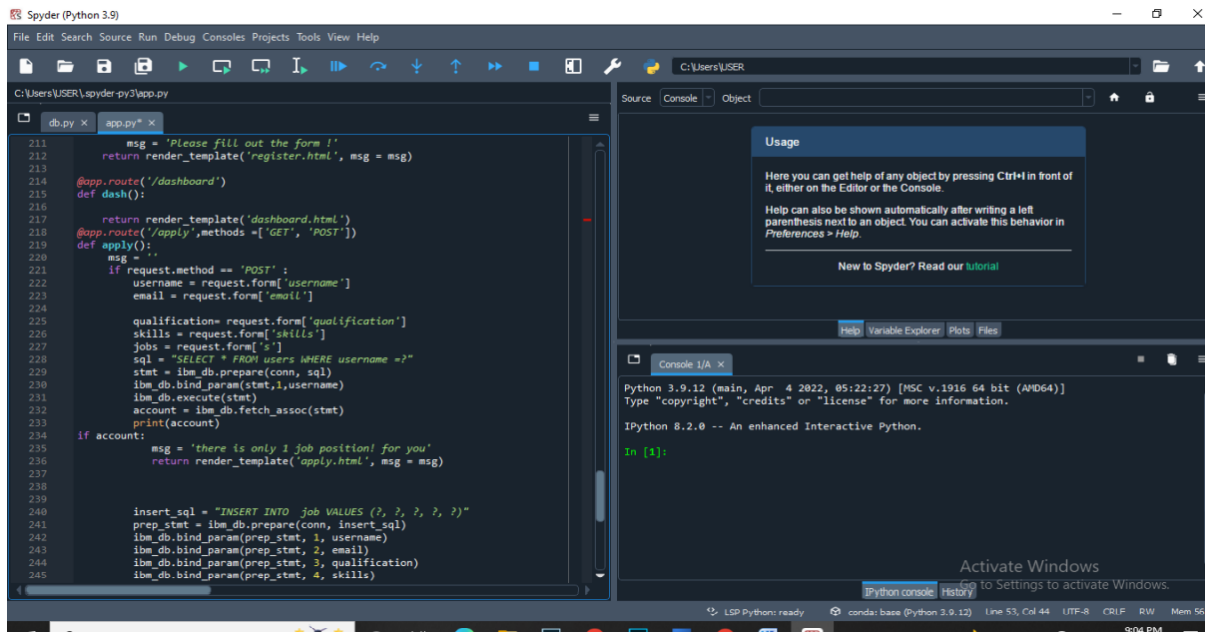
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]:

```





# Register Form

You have successfully registered !

Register

already have an account ? please login [login!](#)

## Apply Now

there is only 1 job position for you

Python

Java

arch



30°C



ENG



18:26

20-10-2022