

## Assignment -2

Assignment Date	26 September 2022
Student Name	R.Manoj
Student Roll Number	311519205037
Maximum Marks	2 Marks

### Question-1:

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

The screenshot shows the IBM Db2 on Cloud console. On the left, a sidebar contains navigation icons. The main area is titled 'Tables' and shows a table named 'USER' under the schema 'PRJ41491'. To the right, the 'Table definition' pane shows the table's structure:

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

Below the table definition, there is a 'View data' button. The top of the console shows various tabs like 'Load Data', 'Load History', 'Tables', 'Views', etc.

- insert into user values('inba@gmail.com','inbatamil','01','a');
- insert into user values('manoj@gmail.com','manoj','02','b');
- insert into user values('aditya@gmail.com','aditya','03','c');

The screenshot shows the IBM Db2 on Cloud console with the 'USER' table data displayed. The table has four columns: EMAIL, USERNAME, ROLL\_NUMBER, and PASSWORD. The data is as follows:

EMAIL	USERNAME	ROLL_NUMBER	PASSWORD
aditya@gmail.com	aditya	03	c
admin@gmail.com	admin	admin	admin
inba@gmail.com	inbatamil	01	a
manoj@gmail.com	manoj	02	b
sabarin4@gmail.com	sabari	sabari	sabari

The interface also shows a 'Back' button and an 'Export to CSV' button. The bottom of the screenshot shows a Windows taskbar with various application icons and system information like '26°C Haze' and '09:00 PM 05-10-2022'.

## Question-2:

Perform UPDATE, DELETE Queries with user table

1.update user set email='sabari@gmail.com' where roll\_number='admin';

The screenshot displays the IBM Db2 on Cloud web interface. The top section shows a SQL editor with the following query:

```
1 update user
2 set email='sabari@gmail.com'
3 where roll_number='admin';
4
5
```

Below the editor is a 'History' table showing the execution of the query:

Script	Date	Status	Runtime
Untitled - 1	Oct 5, 2022 9:03:25 PM	✓ 1	0.040 s
update user set email='sabari@gmail.com' where roll_number='admin'		✓	0.040 s
Untitled - 1	Oct 5, 2022 8:59:58 PM	✓ 3	0.033 s
insert into user values('inba@gmail.com','inbatamil','01','a')		✓	0.010 s
insert into user values('manoj@gmail.com','manoj','02','b')		✓	0.015 s

The bottom section shows the 'CHR84960.USER' table with the following data:

EMAIL	USERNAME	ROLL_NUMBER	PASSWORD
aditya@gmail.com	aditya	03	c
inba@gmail.com	inbatamil	01	a
manoj@gmail.com	manoj	02	b
sabari@gmail.com	admin	admin	admin
sabarin4@gmail.com	sabari	sabari	sabari

## 1. DELETE FROM user WHERE roll\_number='03';

The screenshot displays the IBM Db2 on Cloud web interface. The top section shows a SQL editor with the query `DELETE FROM user WHERE roll_number='03';` entered. Below the editor, a history table lists recent executions. The bottom section shows the 'Data objects' view for the 'CHR84960.USER' table, displaying a list of users with their email, username, roll number, and password.

Script	Date	Status	Runtime
Untitled - 1	Oct 5, 2022 9:06:40 PM	✓ 1	0.048 s
DELETE FROM user WHERE roll_number='03'		✓	0.048 s
Untitled - 1	Oct 5, 2022 9:06:15 PM	✗ 1	0.022 s
DELETE FROM user WHERE roll_number=03		✗	0.022 s
Untitled - 1	Oct 5, 2022 9:06:05 PM	✗ 1	0.057 s

EMAIL	USERNAME	ROLL_NUMBER	PASSWORD
inba@gmail.com	inbatamil	01	a
manoj@gmail.com	manoj	02	b
sabari@gmail.com	admin	admin	admin
sabarin4@gmail.com	sabari	sabari	sabari

### Question 3:

Connect python code to db2

#### Program:

```
from flask import Flask
import ibm_db
```

```

app=Flask(__name__)
try:
    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=prj41491;PWD=8NNe2bnsXzveXAcg","","")
except:
    print("Unable to connect: ",ibm_db.conn_error())

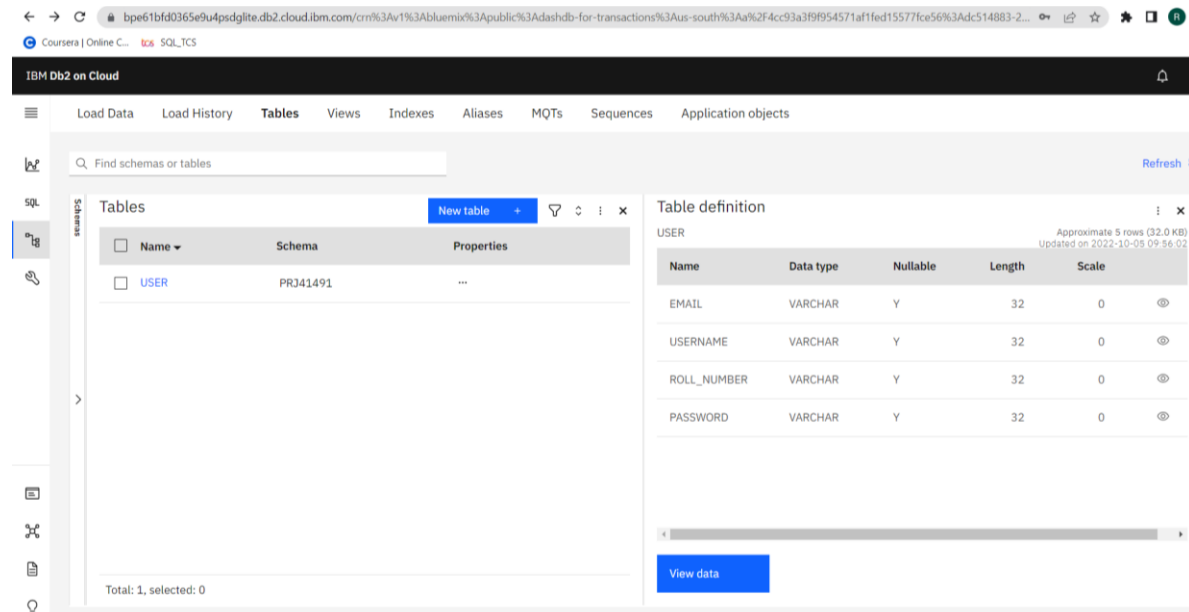
@app.route("/")
def dashboard():
    return "welcome"
if __name__=='__main__':
    app.run(debug=True)

```

#### 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 username and password. If the user is valid show the welcome page.

#### Database:



The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, and a table named 'USER' is listed under the schema 'PRJ41491'. The 'Table definition' panel on the right shows the table structure:

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

The interface also shows a search bar, navigation tabs (Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, Application objects), and a 'View data' button at the bottom right of the table definition panel.

#### Program:

##### User.py

```

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

```

```

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

```

```

try:
    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=prj41491;PWD=8NNe2bnsXzveXAcg","","")
except:
    print("Unable to connect: ",ibm_db.conn_error())

@app.route("/")
def dash():
    return render_template('register.html',msg=" ")

@app.route("/register",methods=['GET','POST'])
def register():
    error = None
    if request.method=='POST':
        username=request.form['username']
        email=request.form['email']
        roll_number=request.form['rollnumber']
        password=request.form['password']
        sql="SELECT * FROM user WHERE roll_number=?"
        prep_stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(prepare_stmt,1,roll_number)
        ibm_db.execute(prepare_stmt)
        account=ibm_db.fetch_assoc(prepare_stmt)
        print(account)
        if account:
            error="Account already exists! Log in to continue !"
        else:
            insert_sql="INSERT INTO user values(?,?,?,?)"
            prep_stmt=ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prepare_stmt,1,email)
            ibm_db.bind_param(prepare_stmt,2,username)
            ibm_db.bind_param(prepare_stmt,3,roll_number)
            ibm_db.bind_param(prepare_stmt,4,password)
            ibm_db.execute(prepare_stmt)
            flash(" Registration successfull. Log in to continue !")
    else:
        pass
    return render_template('login.html',error=error)

@app.route('/login',methods=['GET','POST'])
def login():
    error = None
    if request.method=='POST':
        username=request.form['username']
        password=request.form['password']

```

```

sql="SELECT * FROM user 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']
    session["username"]=account["USERNAME"]
    flash("Logged in successfully!")
    return redirect(url_for("welcome_page"))
else:
    error="Incorrect username / password"
    return render_template('login.html',error=error)

@app.route('/welcome')
def welcome_page():
    return render_template("welcome.html",user=session['id'])

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

```

## register.html

```

<html>
<head>
    <title>Registration page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
    <div class="center"> <br /><br />
    <form action="/register" method="POST">
        <h2>Student Registration Portal</h2> <br /><br />
        <p> Enter Email ID:</p>
        <input type="email" name="email"/>
        <p> Enter Username:</p>
        <input type="text" name="username" />
        <p> Enter Password:</p>
        <input type="password" name="password"/>
        <p> Enter Roll number:</p>
        <input type="text" name="rollnumber"/>
        <br><br><br>
        <input type="submit" value="submit" />
    </form>
</body>
</html>

```

```
        </form>
    </div>
</body>
</html>
```

### login.html

```
<html>
<head>
    <title>Login page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
{% if error %}
    <p><strong style="color:red">Error</strong>: {{error}}</p>
{% endif %}
{% with messages = get_flashed_messages() %}
    {% if messages %}
        {% for message in messages %}
            <p style="color:green">{{ message }}</p>
        {% endfor %}
    {% endif %}
{% endwith %}
<div class="center"> <br /><br />
<form action="/login" method="POST">
<h1>Login</h1> <br /><br />
    <p> Enter Username:</p>
    <input type="text" name="username" />
    <p> Enter Password:</p>
    <input type="password" name="password"/>
    <br><br><br>
    <input type="submit" value="submit" />
</form>
</div>
</body>
</html>
```

### welcome.html

```
<html>
<head>
    <title>Login page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
```

```

{% with messages = get_flashed_messages() %}
    {% if messages %}
        {% for message in messages %}
            <p style="color:green">{{ message }}</p>
        {% endfor %}
    {% endif %}
{% endwith %}

    <br /><br />
    <center>
        <h2> User {{user}}</h2> <br />
        <br /><br />
        <p><i>As long as you keep yourself focused, you will have a mighty mind that can make
everything possible.</i></h3>
    </center><br/>
    <div style="margin:40px">
        <center><h1>Welcome</h1> </center>
    </div>
</body>
</html>

```

### style.css

```

.center{
margin: auto;
width: 15%;
}
h2{
text-align:center;
}
input{
border-radius:5px;
padding:7px;
}

```



## Output:

The image displays two screenshots from a web browser. The top screenshot shows a 'Student Registration Portal' with a registration form. The bottom screenshot shows the 'IBM Db2 on Cloud' console displaying the registration data.

**Student Registration Portal**

Enter Email ID:

Enter Username:

Enter Password:

Enter Roll number:

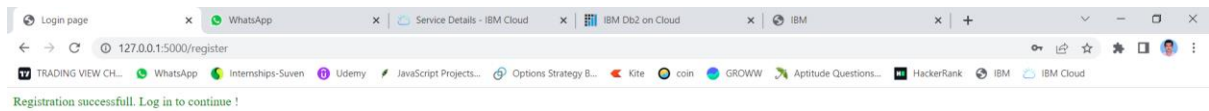
**IBM Db2 on Cloud**

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

CHR84960.USER Back

Export to CSV

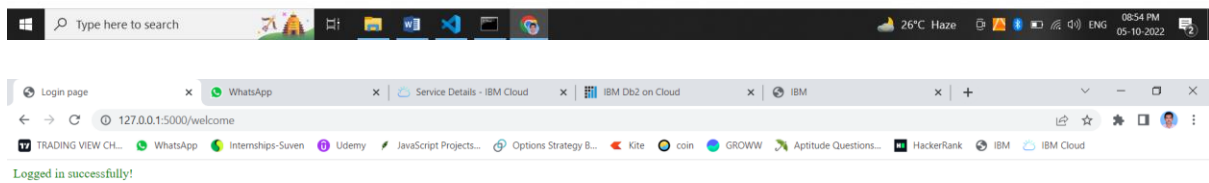
EMAIL	USERNAME	ROLL_NUMBER	PASSWORD
admin@gmail.com	admin	admin	admin
sabarin4@gmail.com	sabari	sabari	sabari



## Login

Enter Username:

Enter Password:



User admin

*As long as you keep yourself focused, you will have a mighty mind that can make everything possible.*

Welcome

