

Assignment -2

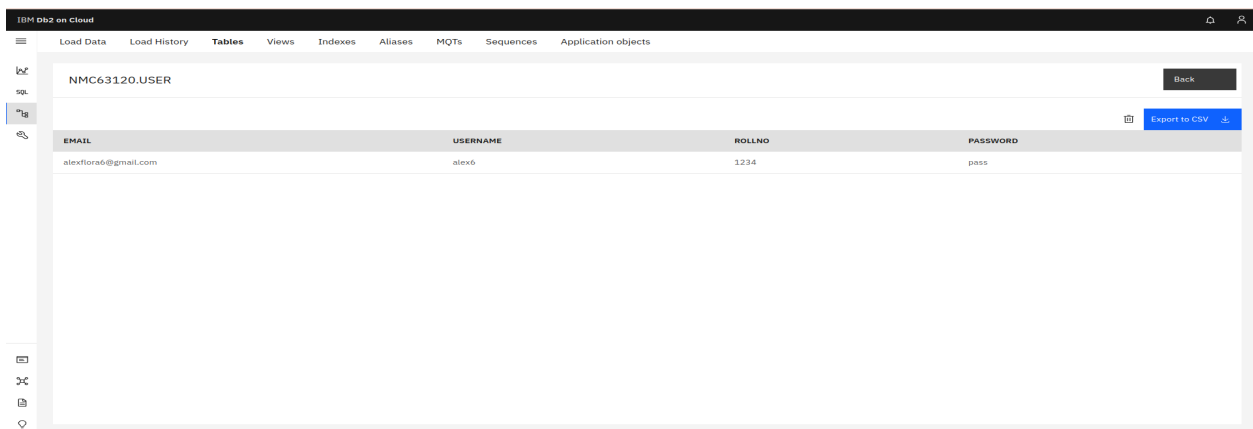
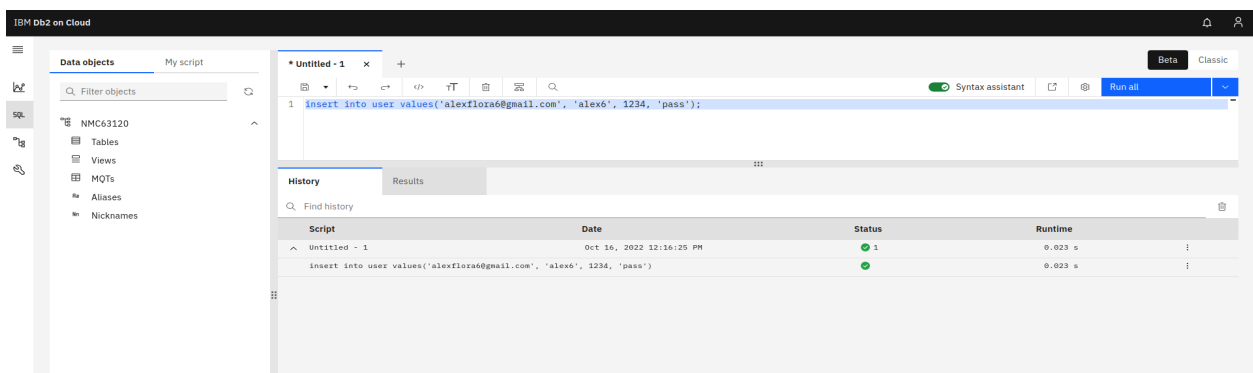
Date	12 October 2022
Student Name	Gowtham E
Project Name	Project -Skill and Job recommender application.

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

A table named 'User' has been created in DB2.

2. Perform UPDATE, DELETE Queries with the User table

Performing insert operation on DB2



Values Inserted!

Updating Values in the user table

IBM Db2 on Cloud

Data objects

My script

Filter objects

NMC63120

Tables

Views

MQTs

Aliases

Nicknames

*Untitled - 1

Syntax assistant

Run all

1

update user set username='no-one' where username='alex6';

History

Results

Find history

Script	Date	Status	Runtime
^ Untitled - 1	Oct 16, 2022 12:21:09 PM	1	0.060 s
update user set username='no-one' where username='alex6'			0.060 s
^ Untitled - 1	Oct 16, 2022 12:16:25 PM	1	0.023 s
insert into user values('alexflora@gmail.com', 'alex6', 1234, 'pass')			0.023 s

IBM Db2 on Cloud

Data objects

My script

Filter objects

NMC63120

Tables

Views

MQTs

Aliases

Nicknames

*Untitled - 1

Syntax assistant

Run from Cursor

1

insert into user values('alexflora@gmail.com', 'alex6', 001, 'pass');

2

insert into user values('guruhan@gmail.com', 'guru', 002, 'pass1');

3

insert into user values('gowtham@gmail.com', 'gowtham', 003, 'pass2');

4

insert into user values('nandhini@gmail.com', 'nandini', 004, 'pass3');

5

update user set username='nandhini' where username='nandini';

6

History

Results

Find history

Script	Date	Status	Runtime
^ Untitled - 1	Oct 16, 2022 12:41:26 PM	1	0.026 s
update user set username='nandhini' where username='nandini'			0.026 s
^ Untitled - 1	Oct 16, 2022 12:35:42 PM	4	0.099 s
insert into user values('alexflora@gmail.com', 'alex6', 001, 'pass')			0.039 s
insert into user values('guruhan@gmail.com', 'guru', 002, 'pass1')			0.020 s
insert into user values('gowtham@gmail.com', 'gowtham', 003, 'pass2')			0.020 s
insert into user values('nandhini@gmail.com', 'nandini', 004, 'pass3')			0.020 s

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

NMC63120.USER

Back

Export to CSV

EMAIL	USERNAME	ROLLNO	PASSWORD
alexflora@gmail.com	alex6	1	pass
gowtham@gmail.com	gowtham	3	pass2
guruhan@gmail.com	guru	2	pass1
nandhini@gmail.com	nandhini	4	pass3

Values are updated in the table.

Deleting a row

IBM Db2 on Cloud

My script

Filter objects

NMC63120

Tables

Views

MQTs

Aliases

Nicknames

Untitled - 1

1

delete from user where rollno=002;

History

Results

Script

Date

Status

Runtime

Untitled - 1

Oct 16, 2022 12:46:23 PM

1

0.019 s

delete from user where rollno=002

1

0.019 s

Untitled - 1

Oct 16, 2022 12:41:26 PM

1

0.026 s

update user set username='nandhini' where username='nandini'

1

0.026 s

Untitled - 1

Oct 16, 2022 12:35:42 PM

4

0.099 s

insert into user values('alexflora6@gmail.com', 'alex6', 'pass')

1

0.039 s

insert into user values('guruharan@gmail.com', 'guru', 'pass1')

1

0.020 s

insert into user values('gowtham@gmail.com', 'gowtham', 'pass2')

1

0.020 s

insert into user values('nandhini@gmail.com', 'nandini', 'pass3')

1

0.020 s

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

NMC63120.USER

Back

Export to CSV

EMAIL

USERNAME

ROLLNO

PASSWORD

alexflora6@gmail.com

alex6

1

pass

gowtham@gmail.com

gowtham

3

pass2

nandhini@gmail.com

nandhini

4

pass3

Row deleted!

3. Connect python code to db2

Program:

first.py

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

```
import ibm_db
```

```
import re
```

```
app=Flask(__name__)
```

```
app.secret_key='a'
```

```
conn =
```

```
ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3  
n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;Security=SSL;SSLServerCert  
ificate=DigiCertGlobalRootCA.crt;UID=nmc63120;PWD=BC9ub7wWLk2Yb8fM;","")
```

```
@app.route('/')
```

```
def home():
```

```
    return render_template("index.html")
```

```
@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'
    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)
        data = ibm_db.execute(stmt)
        if data != True:
            msg="Username Already exists! Please Login"
            print(data)
        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 logged in! from /register'
    return redirect(url_for('dash', username=username))
elif request.method=='POST':
    msg='Please fill out the form'
    return render_template('index.html',msg=msg)

```

```
@app.route('/dashboard')
```

```
def dash():
```

```
    return render_template('dashboard.html',username=request.args.get('username'))
```

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

```
def apply():
```

```
    msg="
```

```
    if request.method=='POST':
```

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

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

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

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

```
        jobs=request.form['s']
```

```
        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="There is only one job position! for you"
    return render_template('apply.html',msg=msg)

```

```

insert_sql="INSERT INTO job 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,qualification)
ibm_db.bind_param(prepare_stmt,4,skills)
ibm_db.bind_param(prepare_stmt,5,jobs)
ibm_db.execute(prepare_stmt)
msg="You have successfully applied for job position"
session["Loggedin"]=True
TEXT = "Hello, a new application for job position"+jobs+"is requested"

```

```

elif request.method=='POST':
    msg="Please fill out the form !"
    return render_template('apply.html',msg=msg)

```

```

@app.route('/display')
def display():
    print(session['username'],session['id'])

```

```

cursor=mysql.connection.cursor()

```

```
cursor.execute('SELECT * FROM job WHERE userid=%s',(session['id'],))
account=cursor.fetchone()
print("accountdisplay",account)
return render_template('display.html',account=account)
```

```
@app.route('/logout')
```

```
def logout():
```

```
    session.pop('Loggedin',None)
    session.pop('id',None)
    session.pop('username',None)
    return render_template('home.html')
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

index.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Document</title>
```

```
    <link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
```

```
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
```

```
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
```

```
integrity="sha384-IDwe1+LCz02ROU9k972gdyvl+AESN10+x7tBKgc9I5HFtuNz0wWnPclzo6p9vxnk" crossorigin="anonymous"></script>
```



```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qb
sw3" crossorigin="anonymous"></script>
```

```
<style>
```

```
.buttons{
display:flex;
color:blue;
width:20%;
justify-content:space-between;
margin-left:40%;
}
```

```
.header{
height:10vw;
background-color:goldenrod;
margin-bottom:20px;
}
```

```
.header>h1{
text-align:center;
padding-top:55px;
}
```

```
.form{
width:75%;
margin:0 auto;
}
```

```
.inputs{
display:flex;
Flex-direction:column;
```

```
}
```

```
.labels{  
display:flex;  
flex-direction:column;  
align-items:flex-end;  
}
```

```
.submit{  
background: transparent;  
border: 1px solid blueviolet;  
border-radius: 10px;  
width:80px;  
}
```

```
.buttons a:hover{  
transition:all .3s;  
background:blueviolet;  
color:white;  
}
```

```
.submit:hover{  
transition:all .3s;  
background:blueviolet;  
color:white;  
}
```

```
.buttons a{  
background: transparent;  
border: 1px solid blueviolet;  
border-radius: 10px;  
text-decoration:none;
```

```
    cursor:pointer;
    width:80px;
    text-align:center;
    color:black;
}
.alert{
    text-align:center;
    align-items:center;
    margin-left:auto;
    margin-right:auto;
}
</style>
</head>
<body>
<div class="container">

    <div class="header">
    <h1>REGISTER</h1>
    </div>
    <div class="form">
        <form action="http://localhost:5000/register" method="post">
            <div class="row">
                <div class="col-3 labels">
                    <label for="email">Email :</label><br><br>
                    <label for="username">Username :</label><br><br>
                    <label for="password">Password :</label><br><br>

                </div>
                <div class="col-9 inputs">
                    <input type="email" name="email" id="email" required><br><br>
```

```
<input type="text" name="username" id="username" required><br><br>
<input type="password" name="password" id="password" required ><br><br>
</div>
</div>
<div class="buttons">
<input class="submit" type="submit" value="Submit">
<a href="http://localhost:5000/login">Login</a>
</div>
</form>
</div>
<div class="alert">{{msg}}</div>
</div>
</body>
</html>
```

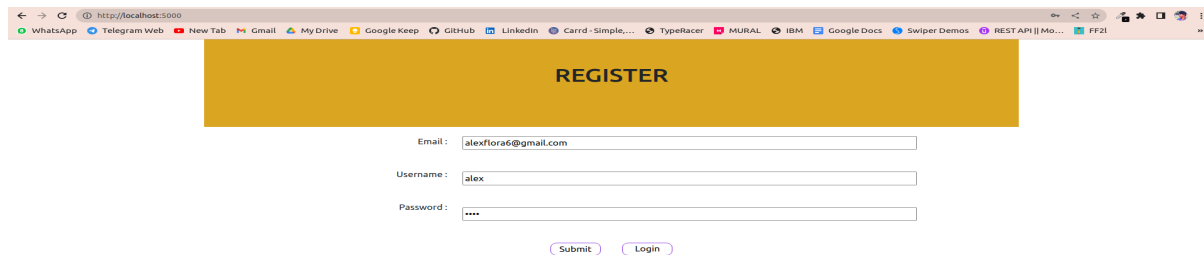
dashboard.html

```
<html>
<head>
<title>LOGIN</title>
</head>
<body>
<div class="container">
<h1>Welcome to dashboard, {{username}}</h1>
</div>
</label>
</form>
</body>
</html>
```

login.html

```
<html>
<head>
<title>LOGIN</title>
</head>
<body>
<form action="http://localhost:5000/login" method="post">
  <label for="username">Username :</label><br>
  <input type="text" name="username" id="username"><br><br>
  <label for="password">Password :</label><br>
  <input type="password" name="password" id="password" ><br><br>
  <input type="submit" value="Submit">
</form>
</label>
</form>
</body>
</html>
```

OUTPUT



The screenshot shows a web browser window with the address bar displaying 'http://localhost:5000'. The browser's tab bar includes several open tabs: WhatsApp, Telegram Web, New Tab, Gmail, My Drive, Google Keep, GitHub, LinkedIn, Carrd - Simple..., Typefacer, MURAL, IBM, Google Docs, Swiper Demos, REST API | Mo..., and FF21. The main content area of the browser displays a registration form. At the top of the form is a yellow header with the word 'REGISTER' in bold black text. Below the header, there are three input fields: 'Email:' with the value 'alexfora6@gmail.com', 'Username:' with the value 'alex', and 'Password:' with masked characters '****'. At the bottom of the form, there are two buttons: 'Submit' and 'Login'.

← → ↻ ⓘ http://localhost:5000/login

🟢 WhatsApp 📠 Telegram Web 📺 New Tab 📧 Gmail 📁 My Drive 📅 Google Keep 🔄 GitHub

Username :

Password :

← → ↻ ⓘ http://localhost:5000/dashboard?username=GOWTHAM

🟢 WhatsApp 📠 Telegram Web 📺 New Tab 📧 Gmail 📁 My Drive 📅 Google Keep 🔄 GitHub

Welcome to dashboard, GOWTHAM

USERNAME	EMAIL	PASSWORD
GOWTHAM	gowtham6@gmail.com	12345678
NANDY	nandy07@gmail.com	123456789
NANDY	gowtham6@gmail.com	12345678
NANDY	nandy07@gmail.com	123456789
hello	alexflora6@gmail.com	alex
hello	ALEXDOOMER@jbj.com	HELLO