

Assignment -2

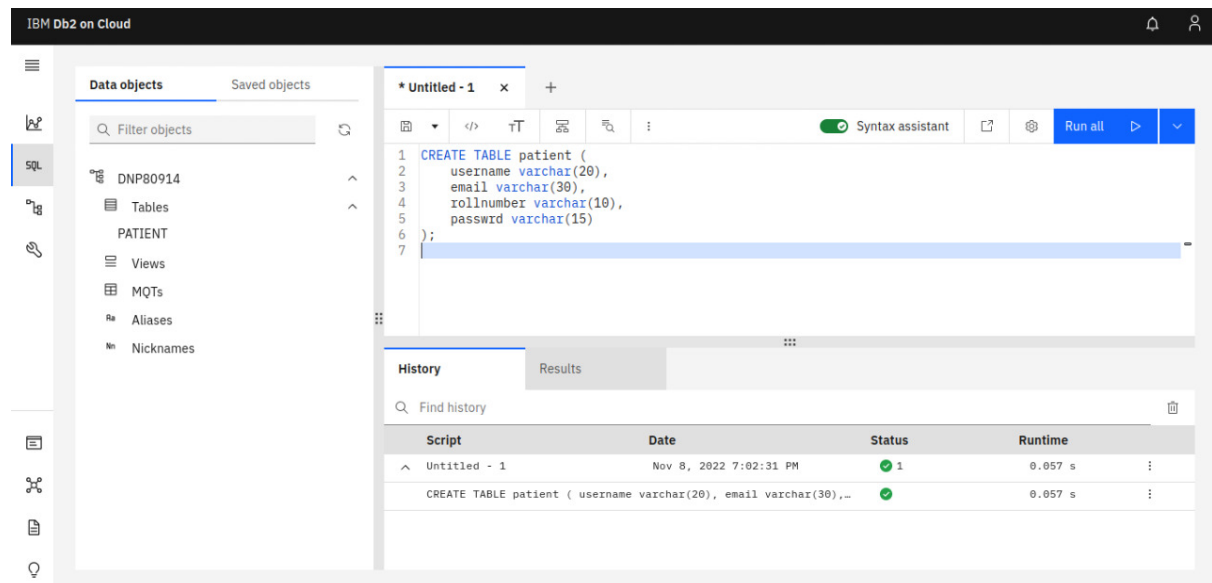
Python Programming

Assignment Date	19 September 2022
Student Name	G.AswinKumar
Student Roll Number	951920LCS05
Maximum Marks	2 Marks

Question-1:

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

```
CREATE TABLE patient (  
    username varchar(20),  
    email varchar(30),  
    rollnumber varchar(10),  
    passwrd varchar(15)  
);
```



Question-2:

2. Perform UPDATE,DELETE Queries with user table

ALTER TABLE

PATIENT

ADD AGE INT;

INSERT

INTO PATIENT

VALUES(

'Aswin',

'aswin@gmail',

'1234',

'adqwagga',

'22'

);

UPDATE

PATIENT

SET

AGE = 20

WHERE

ROLLNUMBER = '1234';

The screenshot shows the IBM Db2 on Cloud interface. On the left, a sidebar displays 'Data objects' and 'Saved objects' with a search bar. Below this, a tree view shows 'DNP80914' containing 'Tables', 'Views', 'MQTs', 'Aliases', and 'Nicknames'. The 'Tables' section is expanded, showing 'PATIENT'. The main area displays a SQL script in a text editor with line numbers 1 to 13. The script is as follows:

```
1 ALTER TABLE
2 PATIENT
3 ADD AGE INT;
4
5 INSERT
6 INTO PATIENT
7 VALUES(
8 'Aswin',
9 'aswin@gmail',
10 '1234',
11 'adqwagga',
12 '22'
13 );
```

Below the script editor, there is a 'History' tab and a 'Results' tab. The 'History' tab is active, showing a table with the following data:

Script	Date	Status	Runtime
Untitled - 1	Nov 8, 2022 7:04:41 PM	✓ 2	0.030 s
ALTER TABLE PATIENT ADD AGE INT		✓	0.022 s
INSERT INTO PATIENT VALUES('Aswin', 'aswin@gmail', '1234', '...		✓	0.008 s

Note: question 3 and 4 result having the same answer

Question-3:

3. Connect python code to db2.

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

import ibm_db

import re

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

conn = ibm_db.connect(
    "DATABASE=bludb;"

    "HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain
    .cloud;"

    "PORT=32733;SECURITY=SSL;"

    "SSLServerCertificate=DigiCertGlobalRootCA.crt;"

    "PROTOCOL=TCPIP;"

    "UID=dnf80914;"

    "PWD=dIWV5iWktjKUDtTY", "", "")

print("Connected to database: ", conn)
print("Connection successful.")

@app.route('/', methods=['POST', 'GET'])
def register():
    msg = "
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        email = request.form['email']
        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)

print(username, password, email)

if account:

    msg = 'Account already exists !'

elif not re.match(r'^[@]+@[^@]+\.[^@]+', email):

    msg = 'Invalid email address !'

    print(msg)

elif not re.match(r'[A-Za-z0-9]+', username):

    msg = 'Username must contain only characters and numbers !'

    print(msg)


elif not username or not password or not email:

    msg = 'Please fill out the form !'

    print(msg)


else:

    insert_sql = 'INSERT INTO users(username, email, password) 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)

    print("User created successfully")

    msg = 'You have successfully registered !'

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


elif request.method == 'POST':

    msg = 'Please fill out the form !'

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

```

```

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

    if request.method == 'GET':
        return render_template('login.html', msg=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)
        print(username, password)

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

            return render_template('home.html', msg=msg, username=username) # Redirect to home
page
        else:
            msg = 'Incorrect username / password !'

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

@app.route('/logout')

```

```
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect(url_for('login'))
```

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

Question-4:

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

home.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Home</title>
  </head>
  <body>
    <h1>Home Page</h1>
    <h2>Welcome {{username}}</h2>
    <p>{{msg}}</p>
    <a href="/logout">Logout</a>
  </body>
</html>
```

login.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Login</title>
  </head>
  <body>
    <h1>LOGIN</h1>
    <p>{{msg}}</p>
    <form action="/login" method="POST">
      <input type="text" name="username" placeholder="Username">
```

```

        <input type="password" name="password" placeholder="Password">
        <input type="submit" value="Login">
    </form>
</body>
</html>

```

registration.html

```

<!DOCTYPE html>
<html>
    <head>
        <title>Register</title>
    </head>
    <body>
        <h1>Register</h1>
        <p>{{msg}}</p>
        <form action="/" method="POST">
            <input type="text" name="username" placeholder="Username">
            <input type="text" name="email" placeholder="Email">
            <input type="password" name="password" placeholder="Password">
            <input type="submit" value="Register">
        </form>
    </body>
</html>

```

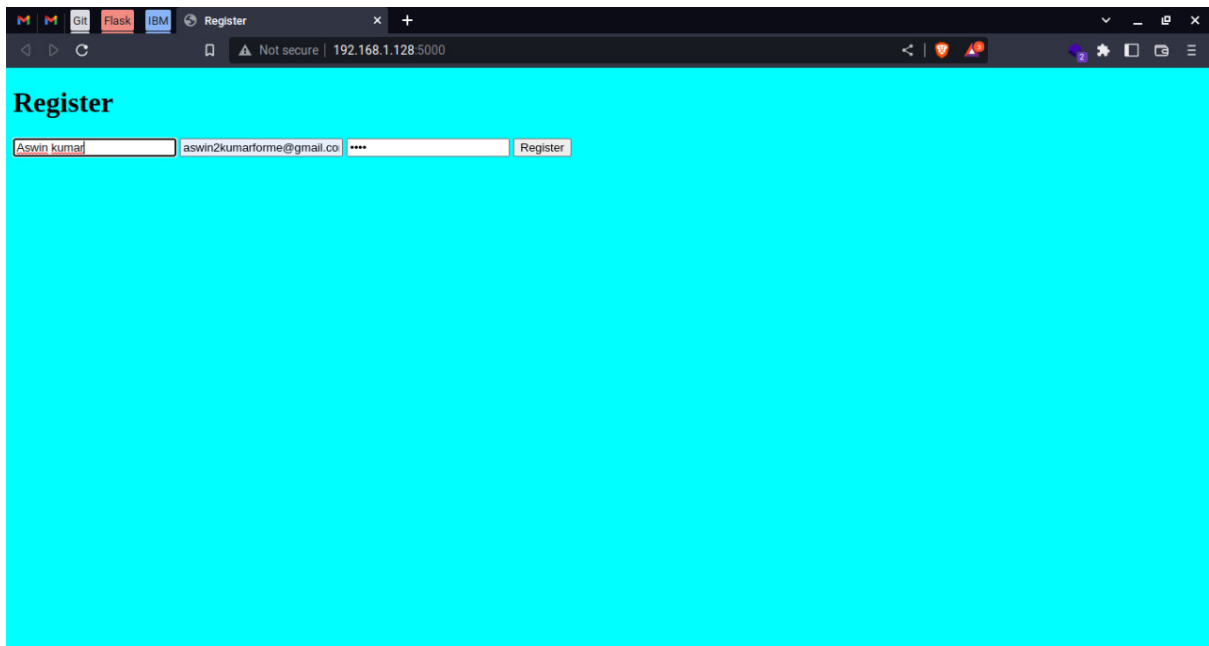
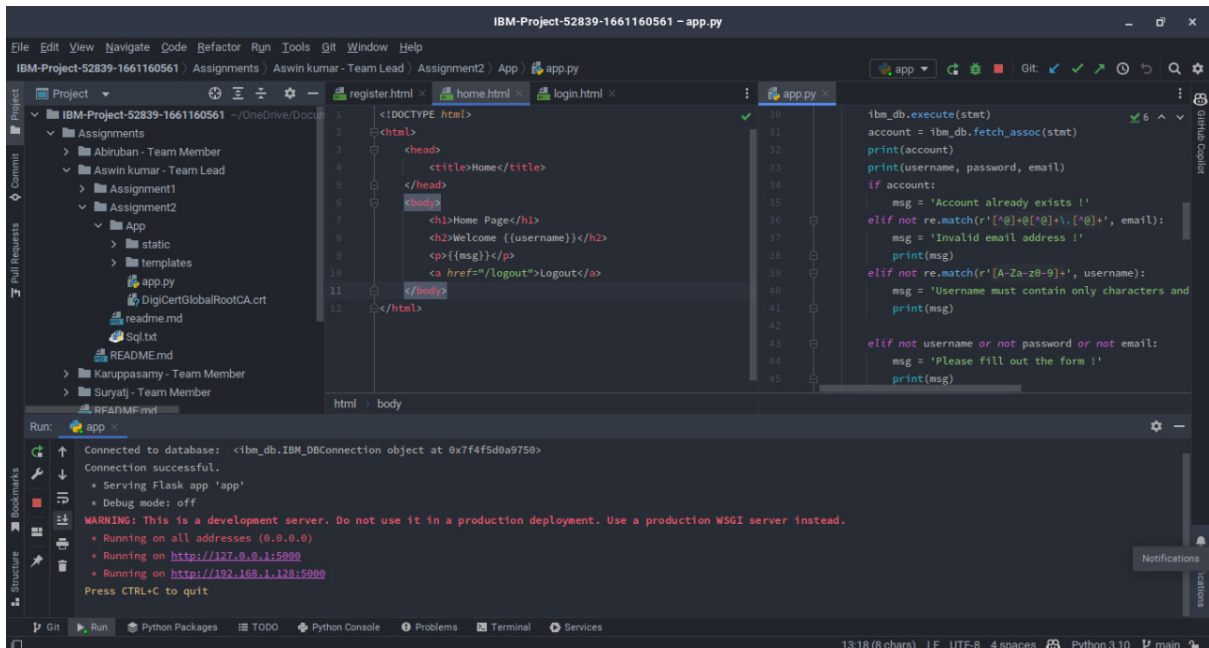
IBM Db2 on Cloud

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

DNP80914.PATIENT Back

Export to CSV ↓

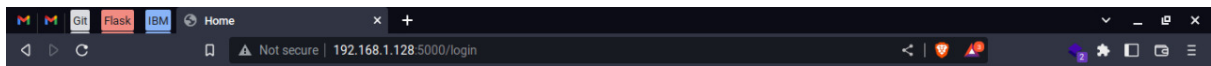
USERNAME	EMAIL	ROLLNUMBER	PASSWRD	AGE
Aswin	aswin@gmail	1234	adqwagga	22





LOGIN

You have successfully registered !



Home Page

Welcome Aswin kumar

Logged in successfully !

[Logout](#)