Assignment -2

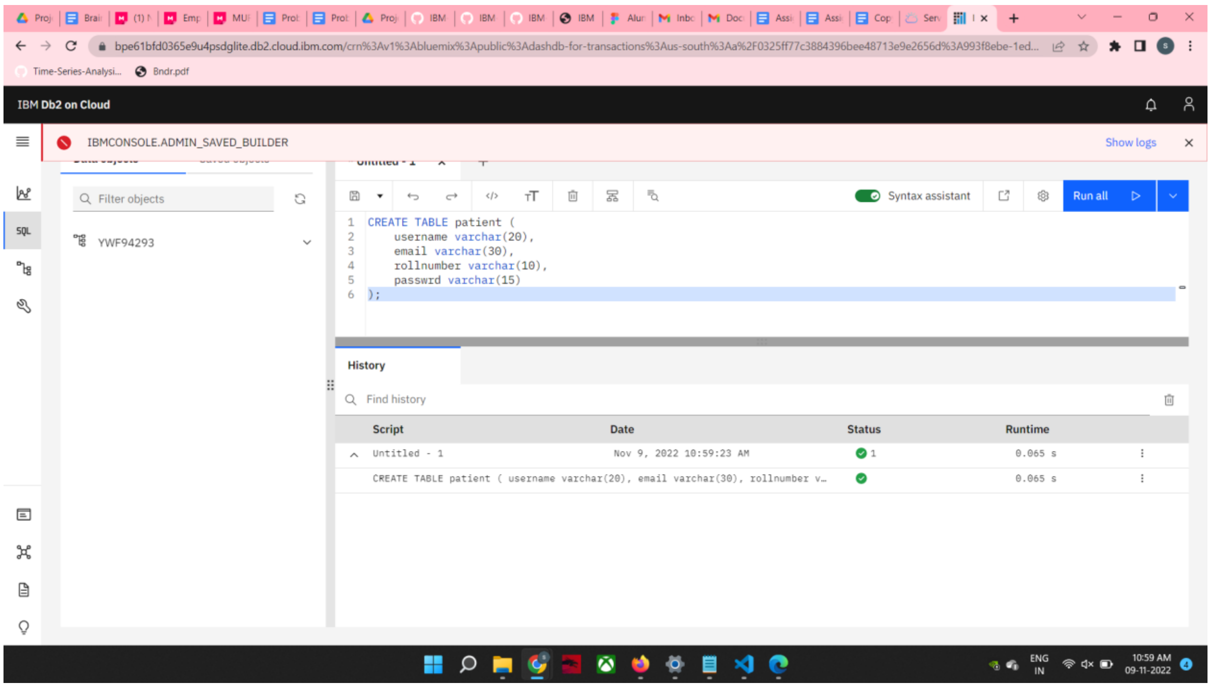
Python Programming

Assignment Date	19 September 2022
Student Name	P.Abiruban
Student Roll Number	951919CS005
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)

);
```



IBM Db2 on Cloud

IBMCONSOLE.ADMIN_SAVED_BUILDER

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

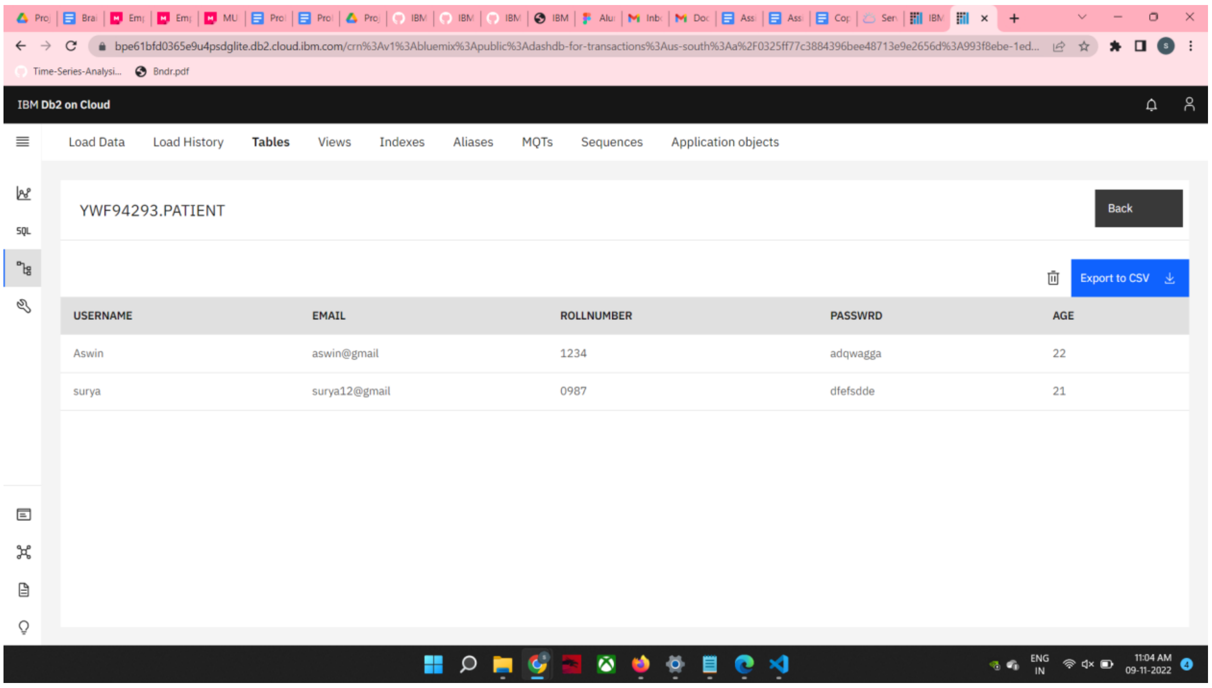
History

	Script	Date	Status	Runtime
^	Untitled - 1	Nov 9, 2022 10:59:23 AM	✓ 1	0.065 s
	CREATE TABLE patient (username varchar(20), email varchar(30), rollnumber v…		✓	0.065 s

Question-2:

2. Perform UPDATE,DELETE Queries with user table

```
ALTER TABLE
  PATIENT
  ADD AGE INT;

INSERT
  INTO PATIENT
  VALUES(
        'surya',
        'surya12@gmail',
        '0987',
        'dfefsdde',
        '21'
);

INSERT
  INTO PATIENT
  VALUES(
        'Aswin',
        'aswin@gmail',
        '1234',
        'adqwagga',
        '22'
);
```



YWF94293.PATIENT

USERNAME	EMAIL	ROLLNUMBER	PASSWRD	AGE
Aswin	aswin@gmail	1234	adqwagga	22
surya	surya12@gmail	0987	dfefsdde	21

<u>Note: question 3 and 4 result having the same answer</u>

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=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;"

"PORT=31249;SECURITY=SSL;"

"SSLServerCertificate=DigiCertGlobalRootCA.crt;"

"UID=djy71776;"

"PWD=cN0mtNWCl6VCBoQ1", "", "")

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(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, password)
            ibm_db.execute(prep_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

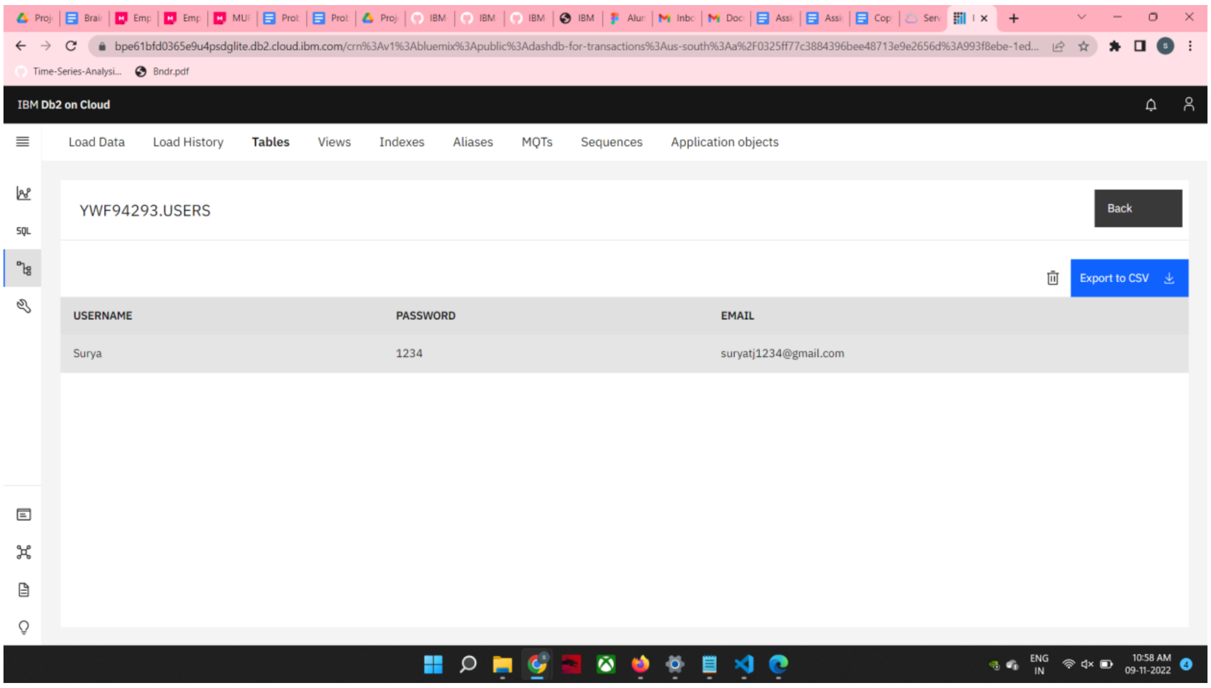
```
<!DOCTYPE html>
<html>
    <head>
        <title>Home</title>
    </head>
    <body>
        <h1>Home Page</h1>
        <h2>Welcome {{username}}</h2>
        <p>{{msg}}</p>
        <a href="/logout">Logout</a>
    </body>

</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

YWF94293.USERS

Back

Export to CSV

USERNAME	PASSWORD	EMAIL
Surya	1234	suryatj1234@gmail.com