

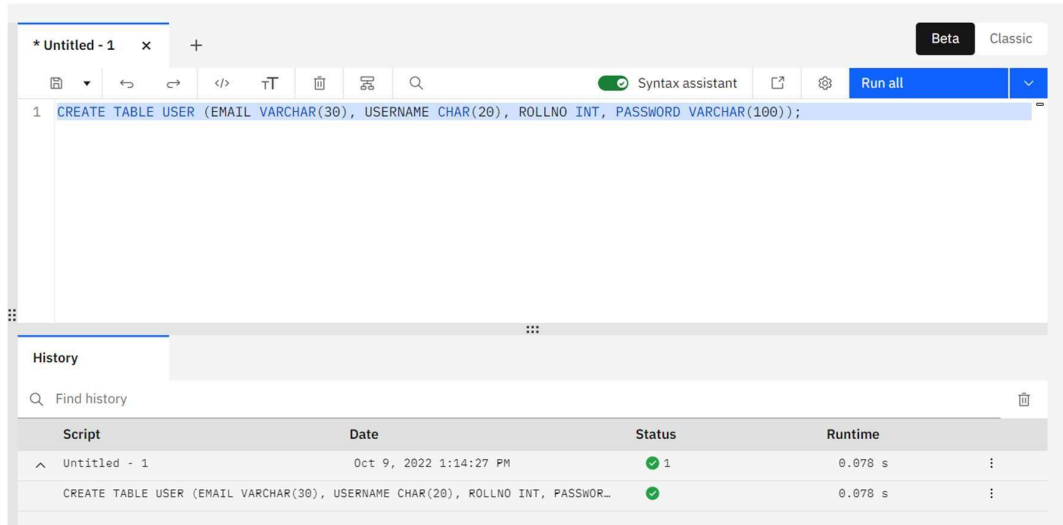
Assignment -2

Connect with Database Assignment

Assignment Date	10 October 2022
Student Name	S.K.Ayngaran
Student Roll Number	621319104008
Maximum Marks	2 Marks

Question-1:

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



2. Perform UPDATE, DELETE Queries with User table

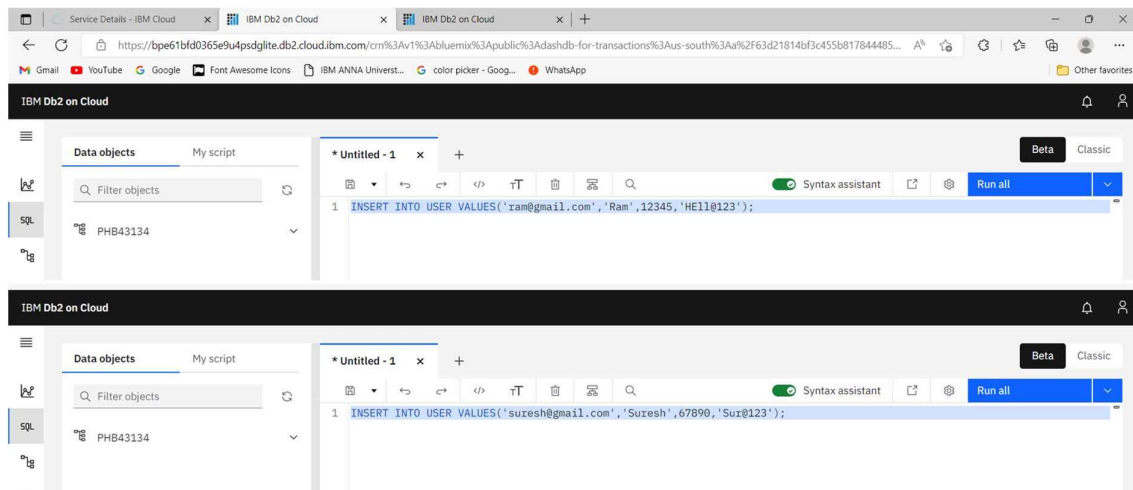


Table View:

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

PHB43134.USER

Back

Export to CSV

EMAIL	USERNAME	ROLLNO	PASSWORD
ram@gmail.com	Ram	12345	HEll@123
suresh@gmail.com	Suresh	67890	Sur@123

UPDATE:

*Untitled - 1 x +BetaClassic

Run all

```
1 UPDATE USER SET ROLLNO=87653 WHERE USERNAME='Ram';
```

Table View:

PHB43134.USER

Back

Export to CSV

EMAIL	USERNAME	ROLLNO	PASSWORD
ram@gmail.com	Ram	87653	HEll@123
suresh@gmail.com	Suresh	67890	Sur@123



DELETE:

*Untitled - 1 x +BetaClassic

Run all

```
1 INSERT INTO USER VALUES('ram@gmail.com','Ram',12345,'HEll@123');
```

TABLE View:

PHB43134.USER				Back
				 Export to CSV 
EMAIL	USERNAME	ROLLNO	PASSWORD	
ram@gmail.com	Ram	87653	HEll@123	

3. Connect python with db2.

Solution:

```
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=grr72787;PWD=527l6pSmLRUddk1Y", "", "")
```

4. Create a flask app with the registration page. Login page and the 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 so the welcome page.

Solution:

app.py

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

```
import ibm_db
```

```
import bcrypt
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=grr72787;PWD=527l6pSmLRUddk1Y", "", "")
```

```
# url_for('static', filename='style.css')
```

```
app = Flask(__name_)
```

```
app.secret_key = 'C21FGSBAPOK43K5VSIDFB2'
```

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

```
def home():
```

```
    if 'email' not in session:
```

```
        return redirect(url_for('login'))
```

```
    return render_template('home.html', name='Home')
```

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

```

def register():
    if request.method == 'POST':
        email = request.form['email']
        username = request.form['username']
        rollNo = request.form['rollNo']
        password = request.form['password']

        if not email or not username or not rollNo or not password:
            return render_template('register.html',error='Please fill all fields')

        hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())

        query = "SELECT * FROM USER WHERE email=? OR rollNo=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,rollNo)
        ibm_db.execute(stmt)
        isUser = ibm_db.fetch_assoc(stmt)

        if not isUser:
            insert_sql = "INSERT INTO User(username,email,PASSWORD,rollNo) 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, hash)
            ibm_db.bind_param(prepare_stmt, 4, rollNo)
            ibm_db.execute(prepare_stmt)
            return render_template('register.html',success="You can login")
        else:
            return render_template('register.html',error='Invalid Credentials')

    return render_template('register.html',name='Home')

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

        if not email or not password:
            return render_template('login.html',error='Please fill all fields')
        query = "SELECT * FROM USER WHERE email=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        isUser = ibm_db.fetch_assoc(stmt)
        print(isUser,password)

```

```

if not isUser:
    return render_template('login.html',error='Invalid Credentials')

isPasswordMatch = bcrypt.checkpw(password.encode('utf-8'),isUser['PASSWORD'].encode('utf-8'))

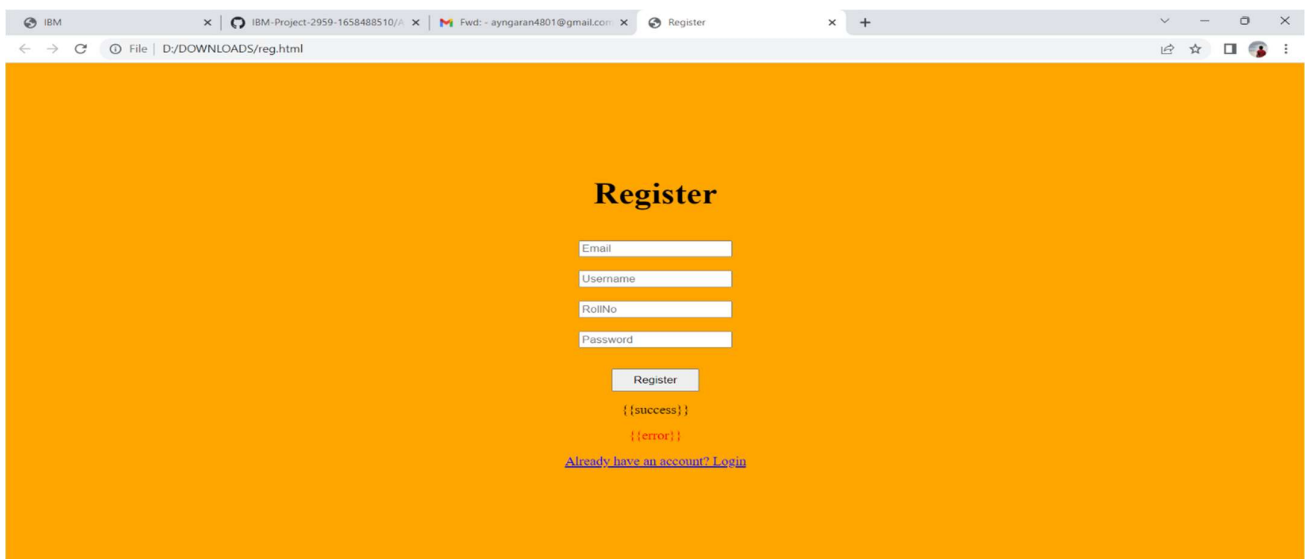
if not isPasswordMatch:
    return render_template('login.html',error='Invalid Credentials')

session['email'] = isUser['EMAIL']
return redirect(url_for('home'))

return render_template('login.html',name='Home')
@app.route('/logout')
def logout():
    session.pop('email', None)
    return redirect(url_for('login'))

```

OUTPUT:



Register

Email

Username

RollNo

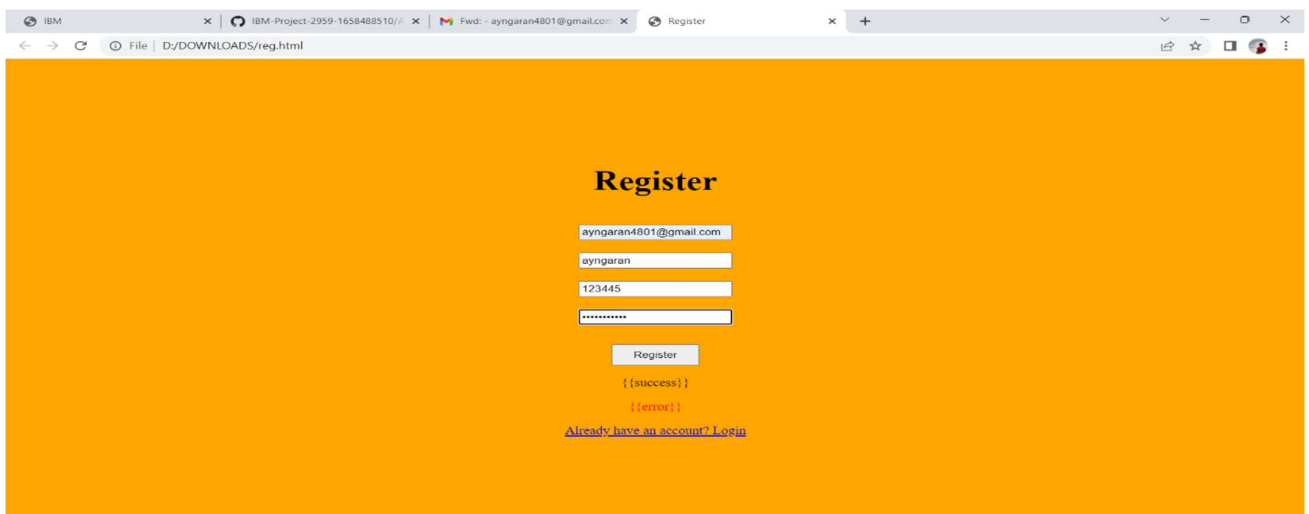
Password

Register

{{success}}

{{error}}

[Already have an account? Login](#)



Register

ayngaran4801@gmail.com

ayngaran

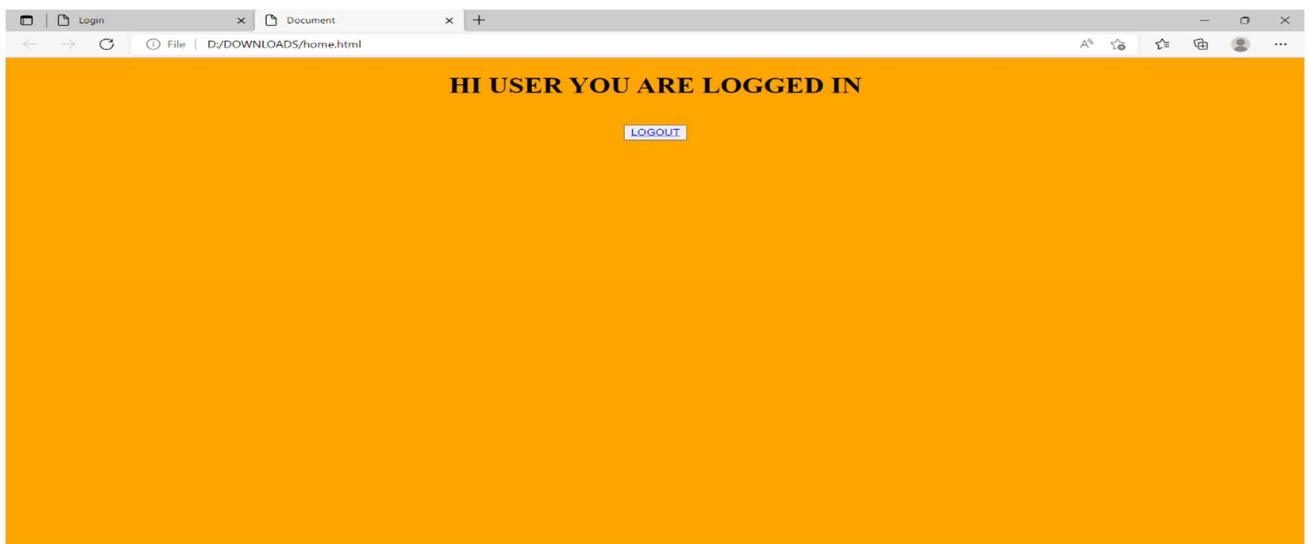
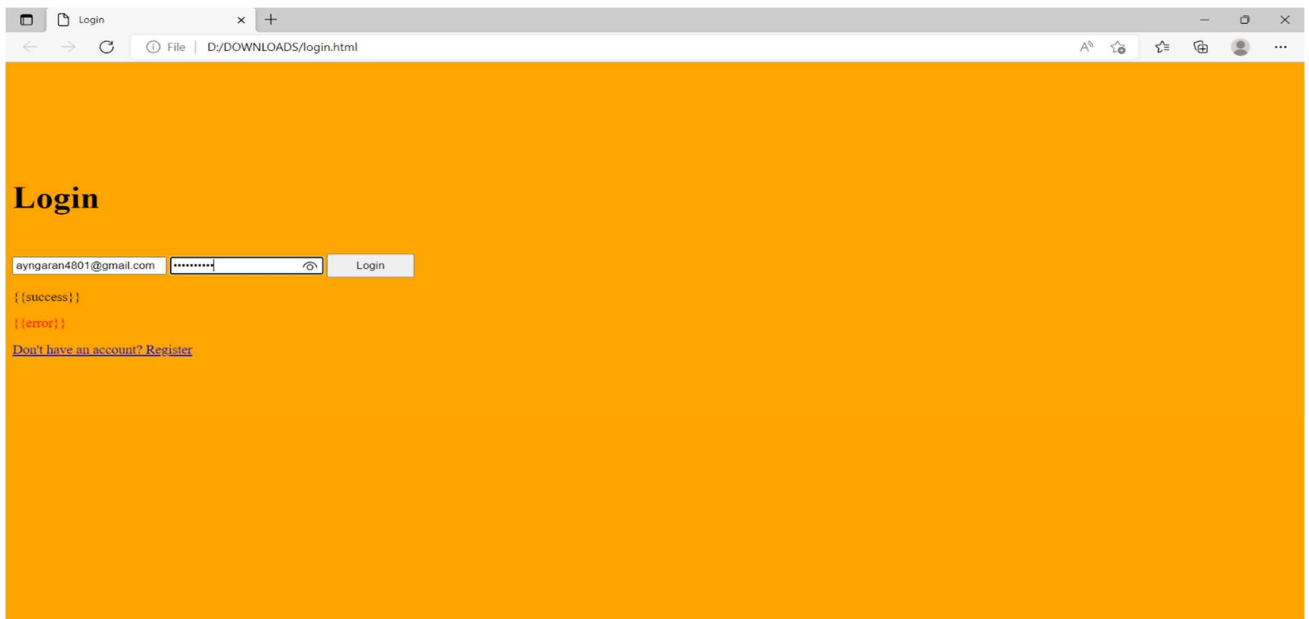
123445

Register

{{success}}

{{error}}

[Already have an account? Login](#)



Database:

2 on Cloud

Load DataLoad History**Tables**ViewsIndexesAliasesMQTsSequencesApplication objects

GRR72787.USER

Back

Export to CSV

EMAIL	USERNAME	PASSWORD	ROLLNO
gfh@gf.v	rt	\$2b\$12\$1U/83.dok7AhdXmpKobXxO/KHFGH7k3Kgb9f6XmKe8EKkPtK4mb22	45