

Assignment -3

templates

home.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>

  </head>

  <body>

    WELCOME

    <button><a href="/logout">LOGOUT</a></button>

  </body>

</html>
```

login.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>Login</title>

  </head>
```

```
<body>

<h3>Login</h3>

<form method="POST">

  <input type="email" name="email" placeholder="Email" required />

  <input type="password" name="password" placeholder="Password" required />

  <button type="submit">Login</button>

</form>

<p>{{ success }}</p>

<p style="color: red">{{ error }}</p>

<a href="/register">Don't have an account? Register</a>

</body>

</html>
```

register.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>Register</title>

</head>

<body class="container">

  <h3>Register</h3>

  <form method="POST">

    <input type="email" name="email" placeholder="Email" required />

    <input type="text" name="username" placeholder="Username" required />

    <input type="text" name="rollNo" placeholder="RollNo" required />
```

```

        <input type="password" name="password" placeholder="Password" required />

        <button type="submit">Register</button>

    </form>

    <p>{{ success }}</p>

    <p style="color: red">{{ error }}</p>

    <a href="/login">Already have an account? Login</a>

</body>

</html>

```

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=;PORT=;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=;PWD=;",",")

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

app = Flask(__name__)

app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'

@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'))

```