ASSIGNMENT 2

ROLL NUMBER	2019503049
NAME	SHIVANI R
TEAM ID	PNT2022TMID35705

1. **Create User table with user with email, username, roll-number, password. Perform UPDATE, DELETE Queries with user table. Connect python code to DB2. 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.**

SOURCE CODE

CREATE STATEMENT

```
CREATE TABLE user (
   username VARCHAR(50) NOT NULL,
   email VARCHAR(50) NOT NULL,
   roll_no VARCHAR(50) NOT NULL,
   password VARCHAR(50) NOT NULL
);
```

INSERT STATEMENT

```
INSERT INTO user
VALUES ('Vignesh Siva P', 'vignesh@gmail.com', '2019503059',
'Password@123');
INSERT INTO user
VALUES ('Ragul B', 'ragul@gmail.com', '2019503036',
'Password@123');
INSERT INTO user
VALUES ('Bharath M', 'bharath@gmail.com', '2019503509',
'Password@123');
INSERT INTO user
VALUES ('Shivani R', 'shivani@gmail.com', '2019503049',
'Password@123');
```

UPDATE STATEMENT

```
UPDATE user SET email = 'vigneshsiva@gmail.com'
where username = 'Vignesh Siva P';
```

DELETE STATEMENT

```
DELETE FROM user
WHERE email = vigneshsiva@gmail.com';
```

CONNECTING PYTHON TO DB2

```
import ibm_db

DATABASE = "bludb"

HOSTNAME = "19af6446-6171-4641-8aba-
9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"

PORT = 30699

UID = "ygg32911"

PWD = "kePH8FKY43ESmASQ"

connection =
ibm_db.connect(f"DATABASE={DATABASE};HOSTNAME={HOSTNAME};PORT={PORT};SECURITY=SSL;S
SLServerCertificate=DigitCertGlobalRootCA.crt;UID={UID};PWD={PWD}", "", "")
```

LOGIN & REGISTER

```
from os import stat

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

import ibm_db

import re


app = Flask(__name__)

app.secret_key = 'Zenik'


DATABASE = "bludb"

HOSTNAME = "fbd88901-ebdb-4a4f-a32e-
9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"

PORT = 32731

UID = "qdp46216"

PWD = "MGhHNGxutNYPFPfE"


connection = ibm_db.connect(
```

```
    f"DATABASE={DATABASE};HOSTNAME={HOSTNAME};PORT={PORT};SECURITY=SSL;SSLServerCertificate=DigitCertGlobalRootCA.crt;UID={UID};PWD={PWD}", "", ""
)

@app.route('/')
@app.route('/home')
def home():
    return render_template('home.html', title='Home', msg=" ")

@app.route('/dashboard')
def dashboard():
    SQL = "SELECT * FROM USERS WHERE username = ?"
    statement = ibm_db.prepare(connection)
    ibm_db.bind_param(statement, 1, session['username'])
    ibm_db.execute(statement)
    account = ibm_db.fetch_assoc(statement)
    return render_template('dashboard.html', title='Dashboard', account=account)

@app.route('/logout')
def logout():
    session.pop('Loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect('/')

@app.route('/login', methods=['GET', 'POST'])
def login():
    message = ""
    global user_id
    if request.method == "POST":
        username = request.form['username']
```

```
        password = request.form['password']
        SQL = "SELECT * FROM USERS WHERE username =? AND password =?"
        statement = ibm_db.prepare(connection)
        ibm_db.bind_param(statement, 1, username)
        ibm_db.bind_param(statement, 2, password)
        ibm_db.execute(statement)
        account = ibm_db.fetch_assoc(statement)
        if account:
            session['Loggedin'] = True
            session['id'] = account['USERNAME']
            session['username'] = account['USERNAME']
            user_id = account['USERNAME']
            return redirect('/dashboard')
        else:
            message = "Incorrect login credentials"
            return render_template('login.html', title='Login', message=message)

@app.route('/register', methods=['GET', 'POST'])
def register():
    message = ""
    if request.method == "POST":
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        roll_number = request.form['roll-number']
        SQL = "SELECT * FROM USERS WHERE username = ? or email= ?"
        statement = ibm_db.prepare(connection)
        ibm_db.bind_param(statement, 1, username)
        ibm_db.bind_param(statement, 2, email)
        ibm_db.execute(statement)
        account = ibm_db.fetch_assoc(statement)
```

```
        if account:
            message = "Account already exists"
        elif not re.match(r'[A-Za-z0-9]+', username):
            message = "Username should be only alphabets and numbers"
        else:
            SQL = "INSERT INTO USERS VALUES (?,?,?,?)"
            statement = ibm_db.prepare(connection)
            ibm_db.bind_param(statement, 1, username)
            ibm_db.bind_param(statement, 2, email)
            ibm_db.bind_param(statement, 3, roll_number)
            ibm_db.bind_param(statement, 4, password)
            ibm_db.execute(statement)
            return redirect('/login')
        return render_template('register.html', message=message, title="Register")

if __name__ == '__main__':
    app.run(debug=True)
```

OUTPUT

USER TABLE

USER

No statistics available.

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	N	50	0
EMAIL	VARCHAR	N	50	0
ROLL_NO	VARCHAR	N	50	0
PASSWORD	VARCHAR	N	50	0

RESULT OF INSERT QUERY

USERNAME	EMAIL	ROLL_NO	PASSWORD
Bharath M	bharath@gmail.com	2019503509	Password@123
Ragul B	ragul@gmail.com	2019503036	Password@123
Shivani R	shivani@gmail.com	2019503049	Password@123
Vignesh Siva P	vignesh@gmail.com	2019503059	Password@123

RESULT OF UPDATE QUERY

USERNAME	EMAIL	ROLL_NO	PASSWORD
Bharath M	bharath@gmail.com	2019503509	Password@123
Ragul B	ragul@gmail.com	2019503036	Password@123
Shivani R	shivani@gmail.com	2019503049	Password@123
Vignesh Siva P	vigneshsiva@gmail.com	2019503059	Password@123

RESULT OF DELETE QUERY

USERNAME	EMAIL	ROLL_NO	PASSWORD
Bharath M	bharath@gmail.com	2019503509	Password@123
Ragul B	ragul@gmail.com	2019503036	Password@123
Shivani R	shivani@gmail.com	2019503049	Password@123

LOGIN & REGISTER