ASSIGNMENT 2

DATE	29th September 2022
ROLL NUMBER	2019503003
NAME	AKSHAYA A

1. **Create registration page in html with username, email and phone number and by using POST method display it in next html page. Develop a flask program which should contain at least 5 packages used from pypi.org.**

   SOURCE CODE

   main.py

```python
from flask import Flask, request, render_template

app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def registerHandler():
    if request.method == "GET":
        return render_template("index.html")

    if request.method == "POST":
        data = {}

        data["name"] = request.form.get("name")
        data["email"] = request.form.get("email")
        data["phone-number"] = request.form.get("phone-number")

        return render_template("success.html", data=data)

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

   templates/index.html

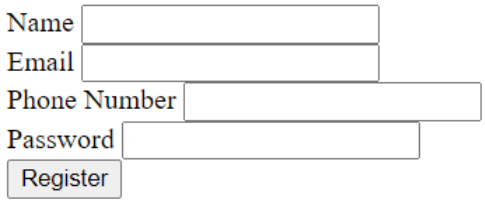
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register</title>
</head>
<body>
```

```
    <div class="wrapper">
        <form method="POST" action="/">
            <div>
                <label for="name">Name</label>
                <input type="text" id="name" name="name" required>
            </div>

            <div>
                <label for="email">Email</label>
                <input type="email" id="email" name="email" required>
            </div>

            <div>
                <label for="phone-number">Phone Number</label>
                <input type="tel" id="phone-number" name="phone-number"
required>
            </div>

            <div>
                <label for="password">Password</label>
                <input type="password" id="password" name="password"
required>
            </div>

            <button type="submit">Register</button>
        </form>
    </div>
</body>
</html>
```

templates/success.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>Success</title>
</head>
<body>
    <div class="wrapper">
        <h1>Registered Successfully</h1>
        <p>Name: {{data['name']}}</p>
        <p>Email: {{data['email']}}</p>
        <p>Phone Number: {{data['phone-number']}}</p>
    </div>
</body>
</html>
```

OUTPUT



2. **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 QDP46216.user (
   username VARCHAR(50) NOT NULL,
   email VARCHAR(50) NOT NULL,
   roll_number VARCHAR(50) NOT NULL,
   password VARCHAR(50) NOT NULL
);
```

INSERT STATEMENT

```
INSERT INTO QDP46216.user
VALUES ('A', 'A@gmail.com', '12345', 'Default@IBM');
```

UPDATE STATEMENT

```
UPDATE QDP46216.user
SET email = 'B@gmail.com'
WHERE username = 'A';
```

DELETE STATEMENT

```
DELETE FROM QDP46216.user
WHERE email = 'B@gmail.com';
```

CONNECTING PYTHON TO DB2

```
import ibm_db
```

```
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;SSLServe
rCertificate=DigitCertGlobalRootCA.crt;UID={UID};PWD={PWD}", "", ""
)
```

LOGIN & REGISTER

```
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;SSLServe
rCertificate=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, SQL)

    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, SQL)

        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, SQL)

        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, SQL)

            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

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

LOGIN & REGISTER

Name
Email
Roll Number
Password
Register

Email
Password
Login