

1. Create User table with user with email, username, roll number, password.
2. Perform UPDATE, DELETE Queries with user table
3. Connect python code to db2.
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

CODEtemplates/index.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>Index</title>
</head>
<body>
  <h2>Welcome to the page</h2>
</body>
</html>
```

templates/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>
  <div>
    <center><h1>User Login</h1></center>
    <form action="/login" method="post">
      <input type="text" name="username" id="username" placeholder="Username"><br>
      <input type="password" name="password" id="password" placeholder="Password"><br>
      <p style="color: red;">{{errorMsg}}</p>
      <p style="color: blue;">{{msg}}</p>
      <input type="submit" value="Submit">
    </form>
  </div>
  <div>
    Not a member? <a href="/register">Regsiter</a>
  </div>
</body>
</html>
```

templates/register.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>User Register</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    input {
      margin-bottom: 20px;
    }
  </style>
</head>

<body>
  <div>
    <center><h1>User Register</h1></center>
    <form action="/register" method="post">
      <input type="text" name="username" id="username" placeholder="Username"><br>
      <input type="password" name="password" id="password" placeholder="Password"><br>
      <input type="email" name="email" id="email" placeholder="Email"><br>
      <input type="text" name="rollno" id="rollno" placeholder="Roll No"><br>
      <input type="submit" value="Submit">
    </form>
  </div>
</body>

</html>
```

app.py

```
import os
import ibm_db
import re
from urllib import request
from flask import Flask, render_template, request, redirect, url_for, session

app = Flask(__name__)

conn = ibm_db.connect(os.environ.get('IBM_DB'), os.environ.get('IBM_DB_USERNAME'),
os.environ.get('IBM_DB_PASSWORD'))

@app.route('/')
def index():
    if session.get("loggedin"):
        msg = 'Logged in successfully !'
        return render_template('index.html', msg=msg)
    else:
        return redirect(url_for("login"))

@app.route('/login', methods=['GET', 'POST'])
def login():
    msg = ""
    if request.method == 'POST':
        username = request.form.get('username')
        password = request.form.get('password')
        sql = "SELECT * FROM Users WHERE username = '?' AND password = '?' limit 1"
```

```

        ibm_db.bind_param(sql, 1, username)
        ibm_db.bind_param(sql, 2, password)
        stmt = ibm_db.exec_immediate(conn, sql)
        result = ibm_db.fetch_assoc(stmt)
        if result != False:
            session['loggedin'] = True
            return redirect(url_for(""))
        msg = "Invalid username/password"
    return render_template('login.html', errorMsg = msg)

```

```

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    return redirect(url_for('login'))

```

```

@app.route('/register', methods=['GET', 'POST'])
def register():
    msg = ""
    errMsg = ""
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form and
    'email' in request.form and 'rollno' in request.form:
        username = request.form.get('username')
        password = request.form.get('password')
        email = request.form.get('email')
        rollno = request.form.get('rollno')

        sql = "SELECT * FROM Users WHERE username = '?"
        ibm_db.bind_param(sql, 1, username)
        stmt = ibm_db.exec_immediate(conn, sql)
        result = ibm_db.fetch_assoc(stmt)
        if result != False:
            errorMsg = 'Account already exists !'
        elif not re.match(r'^@+@[^@]+\.[^@]+', email):
            errorMsg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            errorMsg = 'Username must contain only characters and numbers !'
        elif not username or not password or not email:
            errorMsg = 'Please fill out the form !'
        else:
            sql = "INSERT INTO Users ('username', 'password', 'email', 'rollno') VALUES ('?', '?', '?', '?)"
            ibm_db.bind_param(sql, 1, username)
            ibm_db.bind_param(sql, 2, password)
            ibm_db.bind_param(sql, 3, email)
            ibm_db.bind_param(sql, 4, rollno)
            ibm_db.exec_immediate(conn, sql)
            msg = 'You have successfully registered!'
        elif request.method == 'POST':
            errorMsg = 'Please fill out the form !'
    return render_template('register.html', errorMsg = errorMsg, msg = msg)

```

```

@app.route('/deleteUser', methods=['POST'])
def deleteuser():
    username = request.form.get('username')
    sql = "DELETE FROM Users WHERE username = '?"
    ibm_db.bind_param(sql, 1, username)
    ibm_db.exec_immediate(conn, sql)

```

```

        return 'user deleted'

@app.route('/updateUser', methods=['POST'])
def updateuser():
    username = request.form.get('username')
    password = request.form.get('password')
    email = request.form.get('email')
    rollno = request.form.get('rollno')
    sql = "UPDATE Users set password='?', email='?',rollno='?' WHERE username = '"
    ibm_db.bind_param(sql, 1, password)
    ibm_db.bind_param(sql, 2, email)
    ibm_db.bind_param(sql, 3, rollno)
    ibm_db.bind_param(sql, 4, username)
    ibm_db.exec_immediate(conn, sql)
    return 'User updated'

if __name__ == "__main__":
    app.run(debug=False if os.environ.get('DEBUG') == 'False' else True)

```

OUTPUT



User Register



User Login

Not a member? [Regsiter](#)

Welcome to the page

user deleted

User updated