TEAM ID	PNT2022TMID49561
PROJECT NAME	NUTRITION ASSISTANT
NAME	GANESAN A
ROLL NO	950019104012

1**. Create User table with user with email, username, roll number, password**.
    CREATE TABLE user(email varchar(50) not null,
                                username varchar(20) not null,
                                password varchar(20) not null)

    INSERT INTO user(email, username, password)
                                VALUES('lakz@gmail.com','lakz','12345')

2. **Perform UPDATE,DELETE Queries with user table**

 UPDATE user SET username='ganesan' WHERE
email='gan@gmail.com'
 DELETE FROM user WHERE username = 'lakz'

3. **Connect python code to db2**.

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-
4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;Se
curity=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=srf62404;PW
D=ezin0NepOTwRzFPd;",'','')
```

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.**

App.py:

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

import ibm_db

import re


app = Flask(__name__)

app.secret_key = 'a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;S
ecurity=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=srf62404;PWD=ez
in0NepOTwRzFPd;",'','')


@app.route('/')

@app.route('/login', methods =['GET', 'POST'])

def login():

    msg = ''

    if request.method == 'POST' and 'username' in request.form and 'password' in
request.form:

        username = request.form['username']

        password = request.form['password']
```

```python
        stmt = ibm_db.prepare(conn,'SELECT * FROM users WHERE username = ?
AND password = ?')

        ibm_db.bind_param(stmt,1,username)

        ibm_db.bind_param(stmt,2,password)

        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)

        if account:

            session['loggedin'] = True

            session['username'] = account['USERNAME']

            msg = 'Logged in successfully !'

            return render_template('index.html', msg = msg)

        else:

            msg = 'Incorrect username / password !'

    return render_template('login.html', msg = msg)


@app.route('/logout')

def logout():

    session.pop('loggedin', None)

    session.pop('id', None)

    session.pop('username', None)

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

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

def register():

    msg = ''

    if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form :

        username = request.form['username']

        password = request.form['password']

        email = request.form['email']

        stmt = ibm_db.prepare(conn,'SELECT * FROM users WHERE username = ?')

        ibm_db.bind_param(stmt,1,username)

        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)

        if account:

            msg = 'Account already exists !'

        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):

            msg = 'Invalid email address !'

        elif not re.match(r'[A-Za-z0-9]+', username):

            msg = 'Username must contain only characters and numbers !'

        elif not username or not password or not email:

            msg = 'Please fill out the form !'

        else:

            prep_stmt = ibm_db.prepare(conn,"INSERT INTO users VALUES(?, ?, ?)")
```

```python
            ibm_db.bind_param(prep_stmt, 1, username)

            ibm_db.bind_param(prep_stmt, 2, email)

            ibm_db.bind_param(prep_stmt, 3, password)

            ibm_db.execute(prep_stmt)

            msg = 'You have successfully registered !'

        elif request.method == 'POST':

            msg = 'Please fill out the form !'

        return render_template('register.html', msg = msg)


if __name__ == '__main__':

    app.debug = True

    app.run(host='0.0.0.0',port=8080)
```

INDEX.HTML

```html
<html>

    <head>

            <meta charset="UTF-8">

            <title> Index </title>

            <link rel="stylesheet" href="{{ url_for('static', filename='style.css')
}}">

    </head>

    <body></br></br></br></br></br>
```

```html
            <div align="center">

            <div align="center" class="border">

                    <div class="header">

                            <h1 class="word">Index</h1>

                    </div></br></br></br>

                            <h1 class="bottom">

                                    Hi {{session.username}}!!</br></br> Welcome to
the index page...

                            </h1></br></br></br>

                            <a href="{{ url_for('logout') }}" class="btn">Logout</a>

            </div>

            </div>

        </body>

</html>
```

REGISTER.HTML

```html
<html>

    <head>

            <meta charset="UTF-8">

            <title> Register </title>

            <link rel="stylesheet" href="{{ url_for('static', filename='style.css')
}}">
```

```html
        </head>

        <body></br></br></br></br></br>

            <div align="center">

            <div align="center" class="border">

                <div class="header">

                    <h1 class="word">Register</h1>

                </div></br></br></br>

                <h2 class="word">

                    <form action="{{ url_for('register') }}" method="post">

                    <div class="msg">{{ msg }}</div>

                        <input id="username" name="username"
type="text" placeholder="Enter Your Username" class="textbox"/></br></br>

                        <input id="password" name="password"
type="password" placeholder="Enter Your Password"
class="textbox"/></br></br>

                            <input id="email" name="email" type="text"
placeholder="Enter Your Email ID" class="textbox"/></br></br>

                            <input type="submit" class="btn" value="Sign
Up"></br>

                    </form>

                </h2>

                <p class="bottom">Already have an account? <a
class="bottom" href="{{url_for('login')}}"> Sign In here</a></p>

            </div>
```

```
            </div>

        </body>

</html>
```

LOGIN.HTML

```
<!-- Store this code in 'login.html' file inside the 'templates' folder -->


<html>

    <head>

            <meta charset="UTF-8">

            <title> Login </title>

            <link rel="stylesheet" href="{{ url_for('static', filename='style.css')
}}">

    </head>

    <body></br></br></br></br></br>

        <div align="center">

        <div align="center" class="border">

                <div class="header">

                        <h1 class="word">Login</h1>

                </div></br></br></br>

                <h2 class="word">

                        <form action="{{ url_for('login') }}" method="post">
```

```html
                        <div class="msg">{{ msg }}</div>

                        <input id="username" name="username"
type="text" placeholder="Enter Your Username" class="textbox"/></br></br>

                        <input id="password" name="password"
type="password" placeholder="Enter Your Password"
class="textbox"/></br></br></br>

                        <input type="submit" class="btn" value="Sign
In"></br></br>

                    </form>

                </h2>

                <p class="bottom">Don't have an account? <a class="bottom"
href="{{url_for('register')}}"> Sign Up here</a></p>

            </div>

            </div>

        </body>

</html>
```

STYLE.CSS

```css
.header{

    padding: 0px 120px;

    width: 150px;

    height: 50px;

    background-color: #444844;
```

```css
      border-radius: 20px;

    padding-bottom: 5px;

  }

  .border{

    padding: 80px 50px;

    width: 400px;

    height: 450px;

    border: 1px solid #58238e;

    border-radius: 20px;

    background-color: #373c38;

  }

  .btn {

    padding: 10px 40px;

    background-color: #10161c;

    color: #FFFFFF;

    font-style: oblique;

    font-weight: bold;

    border-radius: 10px;

    padding-bottom: 5px;}

  .textbox{

    padding: 10px 40px;

    background-color: #999d99;
```

```css
    text-color: #FFFFFF;

    border-radius: 10px;

    height: 70px;

    width: 400px;

    font-weight: bold;

    font-size: 15px;

}

    ::placeholder {

    color: #FFFFFF;

    opacity: 1;

    font-style: oblique;

    font-weight: bold;

}

.word{

    color: #FFFFFF;

    font-style: oblique;

    font-weight: bold;

}

.bottom{

    color: #d5dee2;

    font-style: oblique;

    font-weight: bold;
```
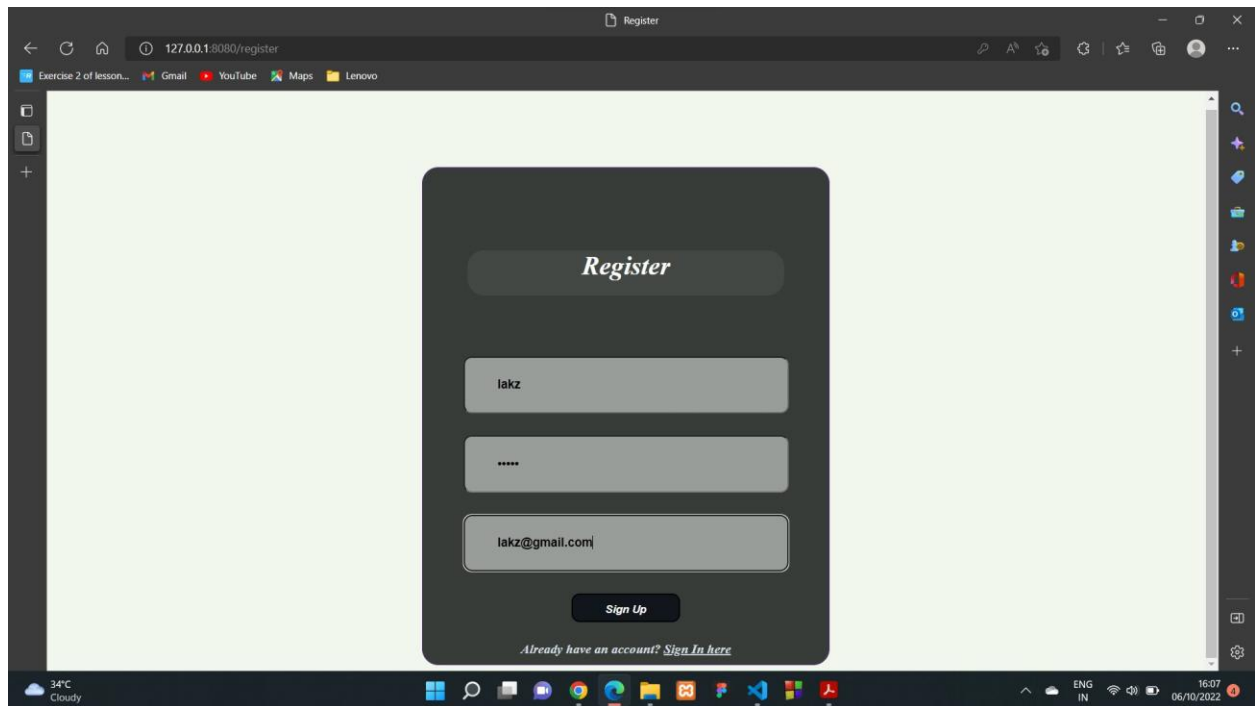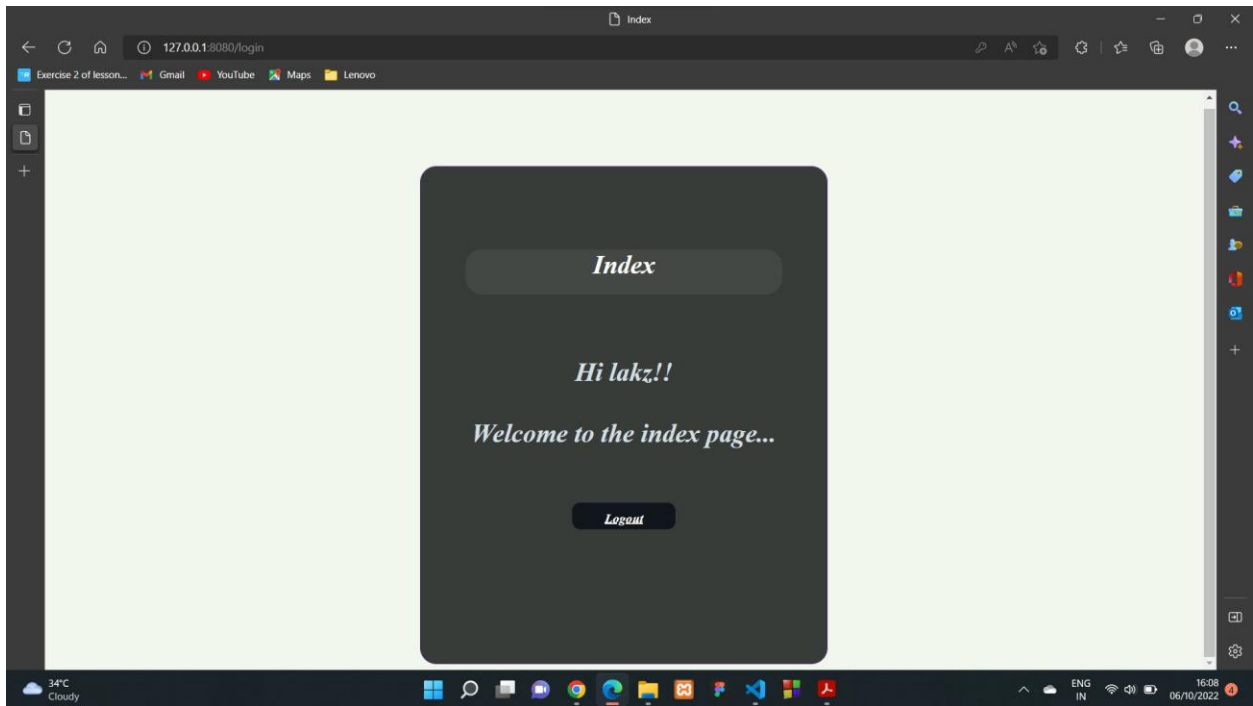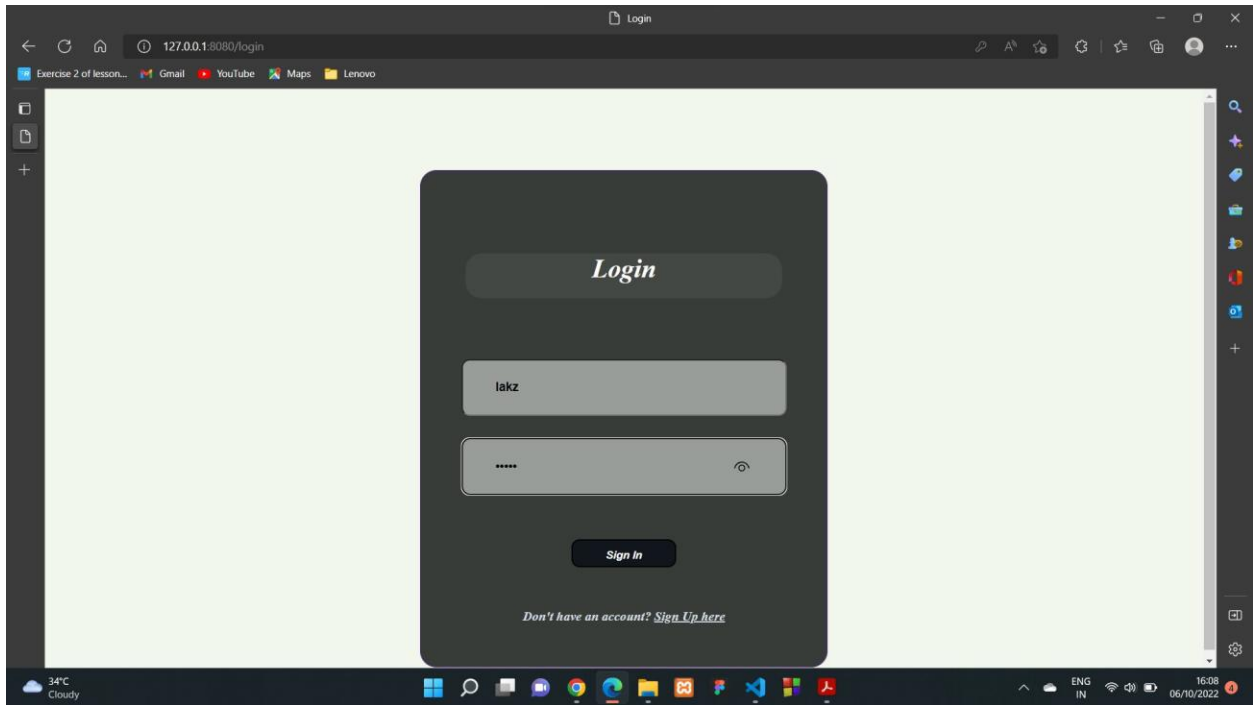
```
}

body{

    background-color: rgba(189, 215, 162, 0.2);

}
```

SCREENSHOTS/OUTPUT

IBM **Db2 on Cloud**

Load Data   Load History   **Tables**   Views   Indexes   Aliases   MQTs   Sequences   Application objects

Q Find schemas or tables

Refresh ↻

Tables

New table   +

	Name ▾	Schema	Properties
☐	USERS	SRF62404	...

Total: 1, selected: 0

Table definition

USERS

Approximate 1 rows (32.0 KB)
Updated on 2022-10-06 08:36:29

Name	Data type	Nullable	Length	Scale	
USERNAME	VARCHAR	N	32	0	◉
EMAIL	VARCHAR	N	51	0	◉
PASSWORD	VARCHAR	N	32	0	◉

View data

IBM **Db2 on Cloud**

Load Data   Load History   **Tables**   Views   Indexes   Aliases   MQTs   Sequences   Application objects

SRF62404.USERS

Back

Export to CSV ⬇

USERNAME	EMAIL	PASSWORD
ganesan	ganesanlakz007@gmail.com	12345
lakz	lakz@gmail.com	12345