

Assignment-2

Team id	PNT2022TMID49642
PROJECT NAME	NUTRITION ASSISTANT APPLICATION
NAME	MOHAMED HAMTHAAN S
ROLL NO	950019106703

1. Create User table with user with email, username, roll number, password.

```
CREATE TABLE user(  
    username varchar(20) not null,  
    email varchar(50) not null,  
    password varchar(20) not null)  
  
INSERT INTO user(username, email, password)  
VALUES('Hamthan', 'hamthan@gmail.com', '12345')
```

2. Perform UPDATE,DELETE Queries with user table

```
UPDATE user SET username='Hamthan' WHERE email= hamthan@gmail.com'DELETE  
FROM user WHERE username = 'Hamthan'
```

3. Connect python code to db2.

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b70af05b-76e4-4bca-a1f5-  
23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;Security=  
SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pbv90190;PWD=pEihUI0HD05igFYt;",'', '')
```

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.

login.html

```
<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">
          <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>
```

register.html:

```
<html>

  <head>

    <meta charset="UTF-8">

    <title> Register </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">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>

```

index.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>
<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>
```

app.py:

```
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=b70af05b-76e4-4bca-a1f5-
23dbb4c6a74e.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;Security=
SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pbv90190;PWD=pEihUI0HD05igFYt;",'', '')

@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']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user 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'))

@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 user 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 user VALUES(?, ?, ?)")
            ibm_db.bind_param(prepare_stmt, 1, username)
```

```
        ibm_db.bind_param(prepare_stmt, 2, email)
        ibm_db.bind_param(prepare_stmt, 3, password)
        ibm_db.execute(prepare_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)
```

Output:

The screenshot displays a web browser window with the address bar showing `127.0.0.1:8080/register`. The browser's tab bar includes several open tabs: IBM, (9) WhatsApp, Service Details - IBM, IBM Db2 on Cloud, IBM, and Register. The Register tab is the active one. The main content area shows a registration form with a light blue background. At the top of the form is a pink button labeled *Register*. Below it are three green input fields containing the text `Hamthan1`, `.....`, and `hamthan12@gmail.com`. A dark blue button labeled *Sign Up* is positioned below the input fields. At the bottom of the form, there is a link that says *Already have an account? [Sign In here](#)*. In the bottom right corner of the browser window, there is a Windows activation watermark that reads "Activate Windows Go to Settings to activate Windows." The Windows taskbar at the bottom of the screen shows the search bar, several application icons, and the system tray with the date and time.

Register

Hamthan1

.....

hamthan12@gmail.com

Sign Up

Already have an account? [Sign In here](#)

Activate Windows
Go to Settings to activate Windows.

Type here to search

32°C Cloudy

ENG IN 2:38 PM 11/9/2022

Register

You have successfully registered !

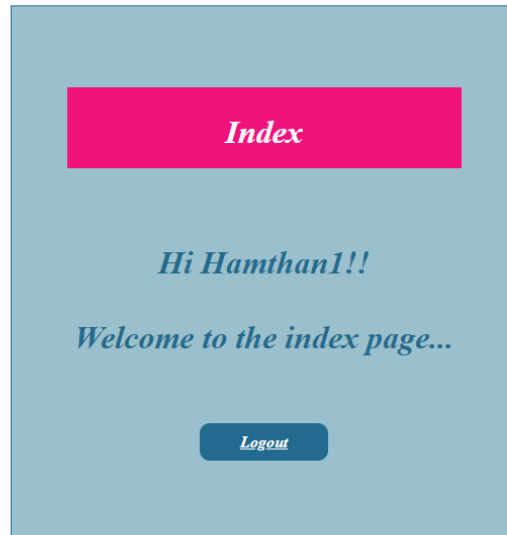
Already have an account? [Sign In here](#)

Activate Windows
Go to Settings to activate Windows.

Login

Don't have an account? [Sign Up here](#)

Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

SQL

PBV90190.USER

Back

Export to CSV

USERNAME	EMAIL	PASSWORD
Hamthan	hamthan@gmail.com	12345
Hamthan1	hamthan12@gmail.com	54321
sumith	asdf@gmail.com	12345

Activate Windows
Go to Settings to activate Windows.

The screenshot displays the IBM Db2 on Cloud console interface. The main area shows the 'Tables' tab with a list of tables. The 'USER' table is selected, and its definition is displayed on the right. The table definition shows three columns: USERNAME (VARCHAR, 20, NULLABLE), EMAIL (VARCHAR, 50, NULLABLE), and PASSWORD (VARCHAR, 20, NULLABLE). The console also shows a sidebar with navigation options like 'Load Data', 'Load History', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'.

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	N	20	0
EMAIL	VARCHAR	N	50	0
PASSWORD	VARCHAR	N	20	0