

SPRINT - 2

DATE	05 November 2022
Team ID	PNT2022TMID50156
Project Name	Project-Nutrition assistant application

CODE:

app.py

```
from turtle import st
from flask import Flask, render_template, request, redirect, url_for, session
from markupsafe import escape

import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30367;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=nbs99713;PWD=s5OJXqYIOlMikOIy",",")

app = Flask(__name__)

@app.route("/")
def index():
    return render_template("index.html")

@app.route("/home")
def home():
    return render_template("home.html")

@app.route("/about")
def about():
    return render_template("about.html")

@app.route("/login")
def login():
    return render_template("login.html")

@app.route("/bmi")
def bmi():
    return render_template("bmi.html")

@app.route("/user")
```

```

def user():
    return render_template("user.html")

@app.route('/addrec', methods = ['POST', 'GET'])
def addrec():
    if request.method == 'POST':

        uname = request.form['uname']
        uemail = request.form['uemail']
        upassword = request.form['upassword']
        conpassword= request.form['conpassword']

        sql = "SELECT * FROM useer WHERE name =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,uname)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template('user.html', msg="You are already a member,
            please login using your details")
        else:
            insert_sql = "INSERT INTO user VALUES (?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, uname)
            ibm_db.bind_param(prepare_stmt, 2, uemail)
            ibm_db.bind_param(prepare_stmt, 3, upassword)
            ibm_db.bind_param(prepare_stmt, 4, conpassword)
            ibm_db.execute(prepare_stmt)

    return render_template('home.html', msg=" user Data saved successfully..")

@app.route('/user')
def user():
    user = []
    sql = "SELECT * FROM user"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        # print ("The Name is : ", dictionary)
        user.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)

    if user:
        return render_template("user.html", user = user)

```

```

@app.route('/delete/<name>')
def delete(name):
    sql = f"SELECT * FROM user WHERE name='{escape(name)}'"
    print(sql)
    stmt = ibm_db.exec_immediate(conn, sql)
    user = ibm_db.fetch_row(stmt)
    print ("The Name is : ", user)
    if user:
        sql = f"DELETE FROM user WHERE name='{escape(name)}'"
        print(sql)
        stmt = ibm_db.exec_immediate(conn, sql)

    user = []
    sql = "SELECT * FROM user"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        user.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
    if user:
        return render_template("user.html", user = user, msg="Delete
        successfully")
        return "success..."

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

```

Index.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>nutrition</title>
    <link rel="stylesheet" href="{ { url_for('static',filename='css/style.css') } }">
    <script src="https://use.fontawesome.com/47d72d0777.js">

    </script>

</head>
<body>
<header class="menu">
    <p>Nutrition &nbsp; Assistant</p>

```

```

        <nav class="nav">
    <ul>
        <li><a href="/home">HOME</a></li>
        <li><a href="/about">ABOUT</a></li>
        <li><a href="/user">LOGIN</a></li>
            <li><a href="/bmi">BMI</a></li>
            <li><a href="/bg">DIET PLAN</a></li>

    </ul>

</nav></header>

<div class="img">
    
</div>
<div class="footer">
<ul>
    <li><a href="/home">HOME</a></li>
    <li><a href="/about">ABOUT</a></li>
    <li><a href="/user">LOGIN</a></li>
</ul>
    <p> &copy; 2022|Designed by VSUT</p>
    <p><a href="https://help.instagram.com/1094643983940381"><i class="fa
fa-instagram" aria-hidden="true"></i></a></p>
    <p><a href="https://twitter.com/login"><i class="fa fa-twitter-square" aria-
hidden="true"></i></a></p>
    <p><a
href="https://www.google.com/account/about/?hl=en&utm_source=google-
account&utm_medium=web"><i class="fa fa-envelope" aria-
hidden="true"></i></a></p>
</div>
</body>
</html>

```

Home.html:

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="stylesheet" href="{ {url_for('static',filename='css/home.css')} }">
    </head>
    <script>
        window.watsonAssistantChatOptions = {

```

integrationID: "8459266a-522b-4340-9ede-c4bfde0f36bf", // The ID of this integration.

region: "eu-gb", // The region your integration is hosted in.

serviceInstanceID: "0077e8b2-9162-48fe-bd46-502637599914", // The ID of your service instance.

```
onLoad: function(instance) { instance.render(); }  
};  
setTimeout(function(){  
    const t=document.createElement('script');  
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/"  
+ (window.watsonAssistantChatOptions.clientVersion || 'latest') +  
"/WatsonAssistantChatEntry.js";  
    document.head.appendChild(t);  
});  
</script>
```

<body >

<body background="{ {url_for('static',filename='images/home2.jpg') }}">

<h1>NUTRITION </h1>

<P>Nutrition is the biochemical and physiological process by which an organism uses food to support its life. It provides organisms with nutrients, which can be metabolized to create energy and chemical structures. Failure to obtain sufficient nutrients causes malnutrition. Nutritional science is the study of nutrition, though it typically emphasizes human nutrition. The type of organism determines what nutrients it needs and how it obtains them. Organisms obtain nutrients by consuming organic matter, consuming inorganic matter, absorbing light, or some combination of these. Some can produce nutrients internally by consuming basic elements, while some must consume other organisms to obtain preexisting nutrients. All forms of life require carbon, energy, and water as well as various other molecules. Animals require complex nutrients such as carbohydrates, lipids, and proteins, obtaining them by consuming other organisms. Humans have developed agriculture and cooking to replace foraging and advance human nutrition. Plants acquire nutrients through soil and the atmosphere. Fungi absorb nutrients around them by breaking them down and absorbing them through the mycelium. </P>

<div class="img">

</div>

<div class="note">

<h2>NUTRITION FACTS</h2>

Added Sugar Is a Disaster

Omega-3 Fats Are Crucial and Most People Don't Get Enough

There Is No Perfect Diet for Everyone

Artificial Trans Fats Are Very Unhealthy

decrease obesity

```
</ol>
</div>
</div>
</body>
```

```
</html>
```

Login.html:

```
<!DOCTYPE html>
<html>
<head>
<title>nutrition</title>
<link rel="stylesheet" href="{ {url_for('static',filename='css/login.css')}}">
</head>
<body>
  <body background="{ {url_for('static',filename='images/black.jpg')}}">

  <div class="header">

    <form>
    <label> ENTER EMAIL </label> <input type="email" required><br><br>
    <label> CONFIRM EMAIL </label> <input type="email" required><br><br>
    <label> COUNTRIES </label>
    <select>
    <option name="COUNTRIES" value="INDIA" >INDIA</option>
    <option name="COUNTRIES" value="USA" >USA</option>
    <option name="COUNTRIES" value="UK" >UK</option>
    <option name="COUNTRIES" value="JAPAN" >JAPAN</option>
    </select>

    <br><br>

    <label> ENTER PASSWORD </label><input type="password" required> <br>
    <br>
    <label> CONFIRM PASSWORD </label><input type="password" required>
    <br>
    <h4><input type="checkbox" name="conditions" value="terms" class="signup">
    yes,i agree to all the terms given by this website.</h4>

    <button type="submit">sign in</button>

    <br><br>
  </form>
```

</div>
</body>
</html>

Dbconnection:

```
OPEN EDITORS 1 unsaved C:\Users> sheeb> Documents> app.py2
app.py
app.py2 C:\Users...
Untitled-1
NUTRITION WEBSITE
static
css
about.css
bg.css
bmi.css
home.css
login.css
read.css
style.css
images
templates
about.html
bg.html
bmi.html
fawesome.html
home.html
index.html
login.html
read.html
regis.html
start.html
user.html
app.py
dbconnectioncode.txt
DigiCertGlobalRootC...
OUTLINE
TIMELINE
```

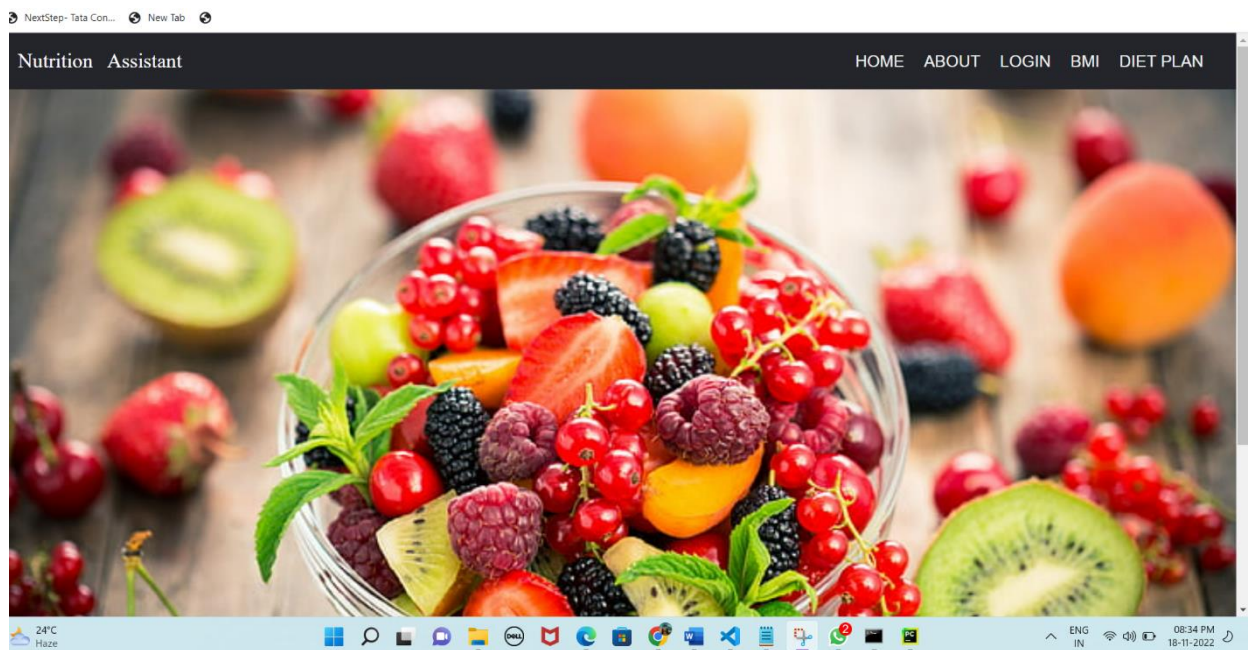
```
1 from turtle import st
2 from flask import Flask, render_template, request, redirect, url_for, session
3 from markupsafe import escape
4
5 import ibm_db
6 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kb1od8lcg.databases.appdomain.cloud;PORT=30367;SECURI
7
8 app = Flask(__name__)
9
10
11 @app.route("/")
12 def index():
13     return render_template("index.html")
14
15 @app.route("/home")
16 def home():
17     return render_template("home.html")
18
19 @app.route("/about")
20 def about():
21     return render_template("about.html")
22
23 @app.route("/login")
24 def login():
25     return render_template("login.html")
26
27 @app.route("/bmi")
28 def bmi():
29     return render_template("bmi.html")
30
31 @app.route("/user")
32 def user():
33     return render_template("user.html")
34
35 @app.route('/addrec', methods = ['POST', 'GET'])
36 def addrec():
37     if request.method == 'POST':
```

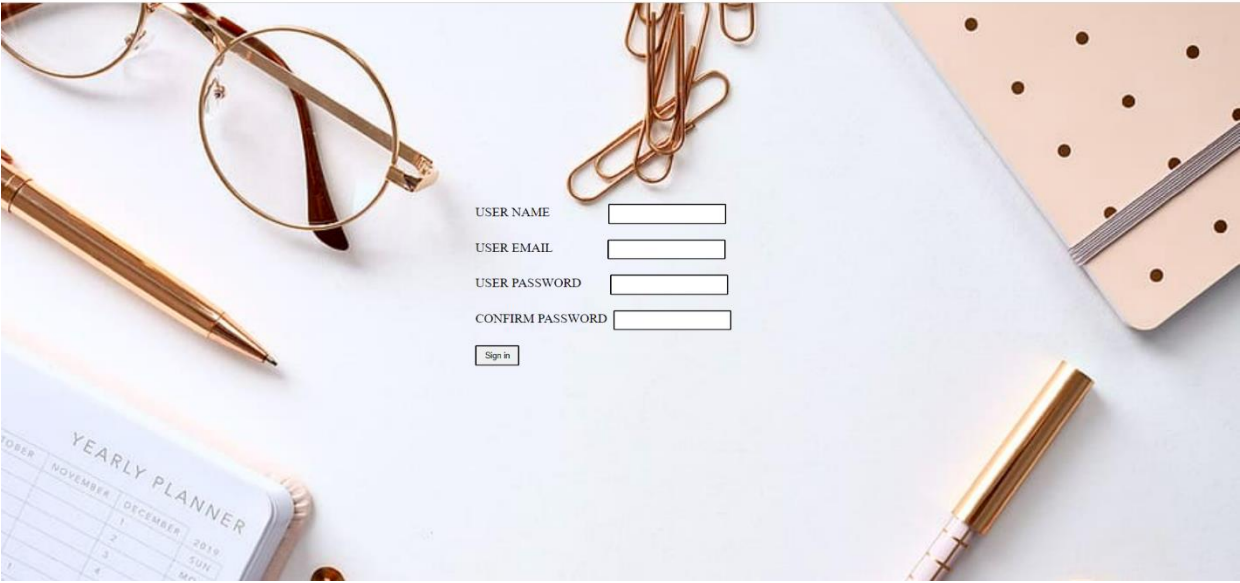
```
OPEN EDITORS 1 unsaved C:\Users> sheeb> Documents> app.py2
app.py
app.py2 C:\Users...
Untitled-1
NUTRITION WEBSITE
static
css
about.css
bg.css
bmi.css
home.css
login.css
read.css
style.css
images
templates
about.html
bg.html
bmi.html
fawesome.html
home.html
index.html
login.html
read.html
regis.html
start.html
user.html
app.py
dbconnectioncode.txt
DigiCertGlobalRootC...
OUTLINE
TIMELINE
```

```
35 @app.route('/addrec', methods = ['POST', 'GET'])
36 def addrec():
37     if request.method == 'POST':
38
39         uname = request.form['uname']
40         uemail = request.form['uemail']
41         upassword = request.form['upassword']
42         compassword= request.form['compassword']
43
44         sql = "SELECT * FROM user WHERE name =?"
45         stmt = ibm_db.prepare(conn, sql)
46         ibm_db.bind_param(stmt,1,uname)
47         ibm_db.execute(stmt)
48         account = ibm_db.fetch_assoc(stmt)
49
50         if account:
51             return render_template('user.html', msg="You are already a member, please login using your details")
52         else:
53             insert_sql = "INSERT INTO user VALUES (?, ?, ?, ?)"
54             prep_stmt = ibm_db.prepare(conn, insert_sql)
55             ibm_db.bind_param(prepare_stmt, 1, uname)
56             ibm_db.bind_param(prepare_stmt, 2, uemail)
57             ibm_db.bind_param(prepare_stmt, 3, upassword)
58             ibm_db.bind_param(prepare_stmt, 4, compassword)
59             ibm_db.execute(prepare_stmt)
60
61             return render_template('home.html', msg=" user Data saved successfully..")
62
63 @app.route('/user')
64 def user():
65     user = []
66     sql = "SELECT * FROM user"
67     stmt = ibm_db.exec_immediate(conn, sql)
68     dictionary = ibm_db.fetch_both(stmt)
69     while dictionary != False:
70         # print ("The Name is : ", dictionary)
71         user.append(dictionary)
```

Ln 41, Col 42 Spaces: 4 UTF-8 CRLF Plain Te

```
OPEN EDITORS 1 unsaved
  app.py
  app.py2 C:\Users\...
  Untitled-1
NUTRITION WEBSITE
  > static
  > templates
  app.py
  dbconnectioncode.txt
  DigiCertGlobalRootC...
C:\Users> sheet > Documents > app.py
70 # print("The Name is : ", dictionary)
71 user.append(dictionary)
72 dictionary = ibm_db.fetch_both(stmt)
73
74 if user:
75     return render_template("user.html", user = user)
76
77 @app.route('/delete/<name>')
78 def delete(name):
79     sql = f"SELECT * FROM user WHERE name='{escape(name)}'"
80     print(sql)
81     stmt = ibm_db.exec_immediate(conn, sql)
82     user = ibm_db.fetch_row(stmt)
83     print("The Name is : ", user)
84     if user:
85         sql = f"DELETE FROM user WHERE name='{escape(name)}'"
86         print(sql)
87         stmt = ibm_db.exec_immediate(conn, sql)
88
89     user = []
90     sql = "SELECT * FROM user"
91     stmt = ibm_db.exec_immediate(conn, sql)
92     dictionary = ibm_db.fetch_both(stmt)
93     while dictionary != False:
94         user.append(dictionary)
95         dictionary = ibm_db.fetch_both(stmt)
96     if user:
97         return render_template("user.html", user = user, msg="Delete successfully")
98     return "success..."
99
100 if __name__ == "__main__":
101     app.run(debug=True)
```





USER NAME

USER EMAIL

USER PASSWORD

CONFIRM PASSWORD

Sign in