

ASSIGNMENT - 2

Date	25 September 2022
Team ID	PNT2022TMID54057
Project Name	Nutrition Assistant Application
Maximum Marks	4 Marks

Questions:

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

The screenshot displays the IBM Db2 on Cloud web interface. The 'Tables' tab is active, showing a list of tables. One table, 'NUTRITABLE', is selected. The 'Table definition' pane on the right provides details for this table:

Name	Data type	Nullable	Length	Scale
ROLLNO	INTEGER	Y	32	0
NAME	CHAR	Y	32	0
EMAIL	CHAR	Y	64	0
PASSWORD	VARCHAR	Y	32	0

A 'View data' button is located at the bottom of the table definition pane.

www.google.com x Service Details - IBM Cloud x IBM Db2 on Cloud x db.pdf.pdf x Document from Tizon - tizon: x +

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F2c4fe265fc2a42c4a1d64b90c8e6c32%3A54b4c6e2-...

IBM Db2 on Cloud

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

LPX03136.NUTRITABLE

Back

Export to CSV

ROLLNO	NAME	EMAIL	PASSWORD
1	TIZON	TIZZ@gmail.com	Lucky@7
1	Jeevi	Mraj@gmail.com	Lucky@27
1	Sargee	Bocthuu@gmail.com	Lucky@77
1	Suji	Bujji@gmail.com	Lucky@8
1	Pradeepa	Tizon@gmail.com	Lucky@7

Type here to search

29°C Haze 21:31 01-11-2022

www.google.com x Service Details - IBM Cloud x IBM Db2 on Cloud x db.pdf.pdf x Document from Tizon - tizon: x +

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F2c4fe265fc2a42c4a1d64b90c8e6c32%3A54b4c6e2-...

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

LPX03136

NUTRITABLE

Views

MQTs

Aliases

Nicknames

*Untitled - 1 x +

Syntax assistant Run all

```
1 INSERT INTO NUTRITABLE VALUES(1,'Pradeepa','Tizon@gmail.com','Lucky@7');
2 INSERT INTO NUTRITABLE VALUES(1,'Suji','Bujji@gmail.com','Lucky@8');
3 INSERT INTO NUTRITABLE VALUES(1,'Sargee','Bocthuu@gmail.com','Lucky@77');
4 INSERT INTO NUTRITABLE VALUES(1,'Jeevi','Mraj@gmail.com','Lucky@27');
5
```

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 1, 2022 9:28:43 PM	4	0.026 s
INSERT INTO NUTRITABLE VALUES(1,'Pradeepa','Tizon@gmail.com','Lucky@7')		✓	0.008 s
INSERT INTO NUTRITABLE VALUES(1,'Suji','Bujji@gmail.com','Lucky@8')		✓	0.006 s
INSERT INTO NUTRITABLE VALUES(1,'Sargee','Bocthuu@gmail.com','Lucky@77')		✓	0.006 s
INSERT INTO NUTRITABLE VALUES(1,'Jeevi','Mraj@gmail.com','Lucky@27')		✓	0.006 s
Untitled - 1	Nov 1, 2022 9:26:48 PM	1	0.006 s
INSERT INTO NUTRITABLE VALUES(1,'TIZON','TIZZ@gmail.com','Lucky@7')		✓	0.006 s

Type here to search

29°C Haze 21:30 01-11-2022

IBM Db2 on Cloud

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

LPX03136.NUTRITABLE Back

Export to CSV

ROLLNO	NAME	EMAIL	PASSWORD
1	TIZON	TIZZ@gmail.com	Lucky@7
1	Jeevi	Mraj@gmail.com	Lucky@27
1	Sargee	Boothuu@gmail.com	Lucky@77
1	Suji	Bujji@gmail.com	Lucky@8
1	Pradeepa	Tizon@gmail.com	Lucky@7

2. Perform UPDATE,DELETE Queries with user table

UPDATE :

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

LPX03136

Tables Views MQTs Aliases Nicknames

*Untitled - 1

```

1 UPDATE NUTRITABLE SET Rollno='2' WHERE Name='Suji';
2 UPDATE NUTRITABLE SET Rollno='3' WHERE Name='Sargee';
3 UPDATE NUTRITABLE SET Rollno='4' WHERE Name='Jeevi';

```

Syntax assistant Run all

History Results

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 1, 2022 9:34:25 PM	3	0.027 s
UPDATE NUTRITABLE SET Rollno='2' WHERE Name='Suji'			0.011 s
UPDATE NUTRITABLE SET Rollno='3' WHERE Name='Sargee'			0.008 s
UPDATE NUTRITABLE SET Rollno='4' WHERE Name='Jeevi'			0.008 s
Untitled - 1	Nov 1, 2022 9:33:12 PM	1	0.008 s
UPDATE NUTRITABLE SET Rollno='2' WHERE Name='Suji'			0.008 s
Untitled - 1	Nov 1, 2022 9:32:21 PM	1	0.005 s
UPDATE NUTRITABLE SET Rollno='2' WHERE Name='suji'			0.005 s
Untitled - 1	Nov 1, 2022 9:32:01 PM	1	0.008 s
UPDATE NUTRITABLE SET Rollno=2 WHERE Name='suji'			0.008 s

The screenshot shows the IBM Db2 on Cloud console interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected, displaying a table named 'LPX03136.NUTRITABLE'. A 'Back' button is in the top right corner. Below the table name, there is an 'Export to CSV' button. The table data is as follows:

ROLLNO	NAME	EMAIL	PASSWORD
1	TIZON	TIZZ@gmail.com	Lucky@7
1	Pradeepa	Tizon@gmail.com	Lucky@7
2	Suji	Bujji@gmail.com	Lucky@8
3	Sargee	Boothuu@gmail.com	Lucky@77
4	Jeevi	Mraj@gmail.com	Lucky@27

DELETE :

The screenshot shows the IBM Db2 on Cloud console interface with the 'Data objects' tab selected. The 'Tables' section is expanded, showing 'LPX03136'. The 'Tables' tab is selected, displaying a table named 'LPX03136.NUTRITABLE'. A 'Back' button is in the top right corner. Below the table name, there is an 'Export to CSV' button. The table data is as follows:

ROLLNO	NAME	EMAIL	PASSWORD
1	TIZON	TIZZ@gmail.com	Lucky@7
1	Pradeepa	Tizon@gmail.com	Lucky@7
2	Suji	Bujji@gmail.com	Lucky@8
3	Sargee	Boothuu@gmail.com	Lucky@77
4	Jeevi	Mraj@gmail.com	Lucky@27

The 'History' tab is selected, showing a list of executed SQL statements. The first statement is a DELETE statement:

```
DELETE FROM NUTRITABLE WHERE Name='TIZON';
```

The execution history table is as follows:

Script	Date	Status	Runtime
Untitled - 1	Nov 1, 2022 9:36:19 PM	✓ 1	0.007 s
DELETE FROM NUTRITABLE WHERE Name='TIZON'		✓	0.007 s
Untitled - 1	Nov 1, 2022 9:34:25 PM	✓ 3	0.027 s
UPDATE NUTRITABLE SET Rollno='2' WHERE Name='Suji'		✓	0.011 s
UPDATE NUTRITABLE SET Rollno='3' WHERE Name='Sargee'		✓	0.008 s
UPDATE NUTRITABLE SET Rollno='4' WHERE Name='Jeevi'		✓	0.008 s
Untitled - 1	Nov 1, 2022 9:33:12 PM	✓ 1	0.008 s
UPDATE NUTRITABLE SET Rollno='2' WHERE Name='Suji'		✓	0.008 s
Untitled - 1	Nov 1, 2022 9:32:21 PM	⚠ 1	0.005 s
UPDATE NUTRITABLE SET Rollno='2' WHERE Name='suji'		⚠	0.005 s

IBM Db2 on Cloud

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

LPX03136.NUTRITABLE Back

Export to CSV

ROLLNO	NAME	EMAIL	PASSWORD
1	Pradeepa	Tizon@gmail.com	Lucky@7
2	Suji	Buji@gmail.com	Lucky@8
3	Sargee	Boothuu@gmail.com	Lucky@77
4	Jeevi	Mrj@gmail.com	Lucky@27

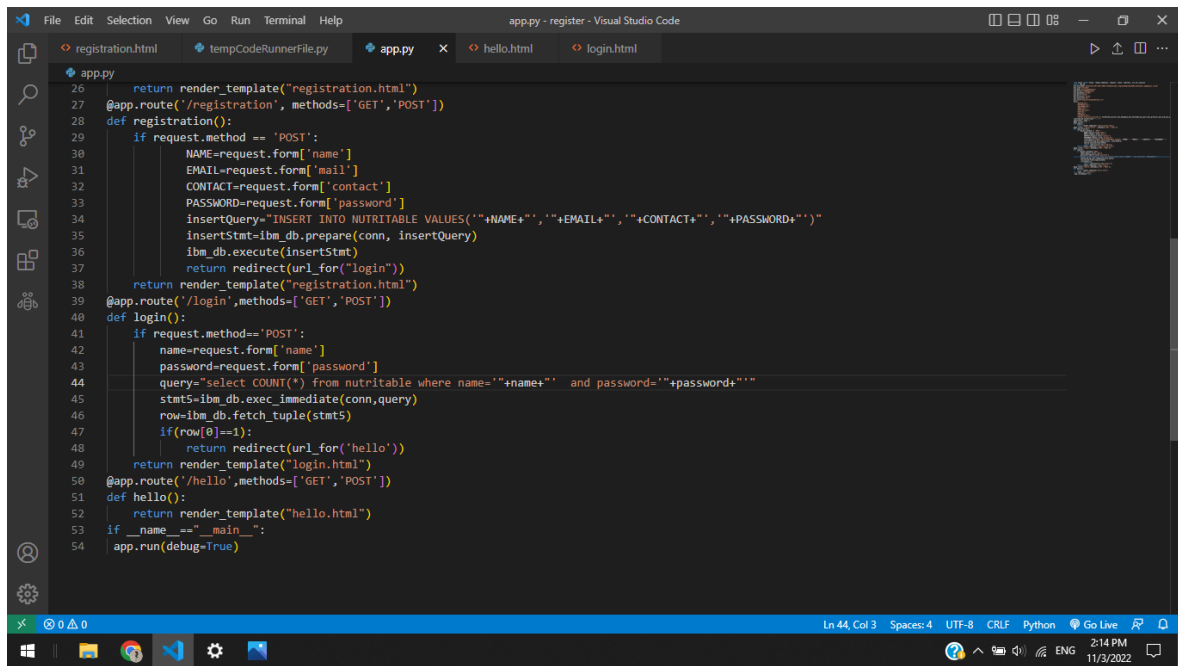
3. Connect python code to db2.
4. Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid show the welcome page.

Registration.html :

```

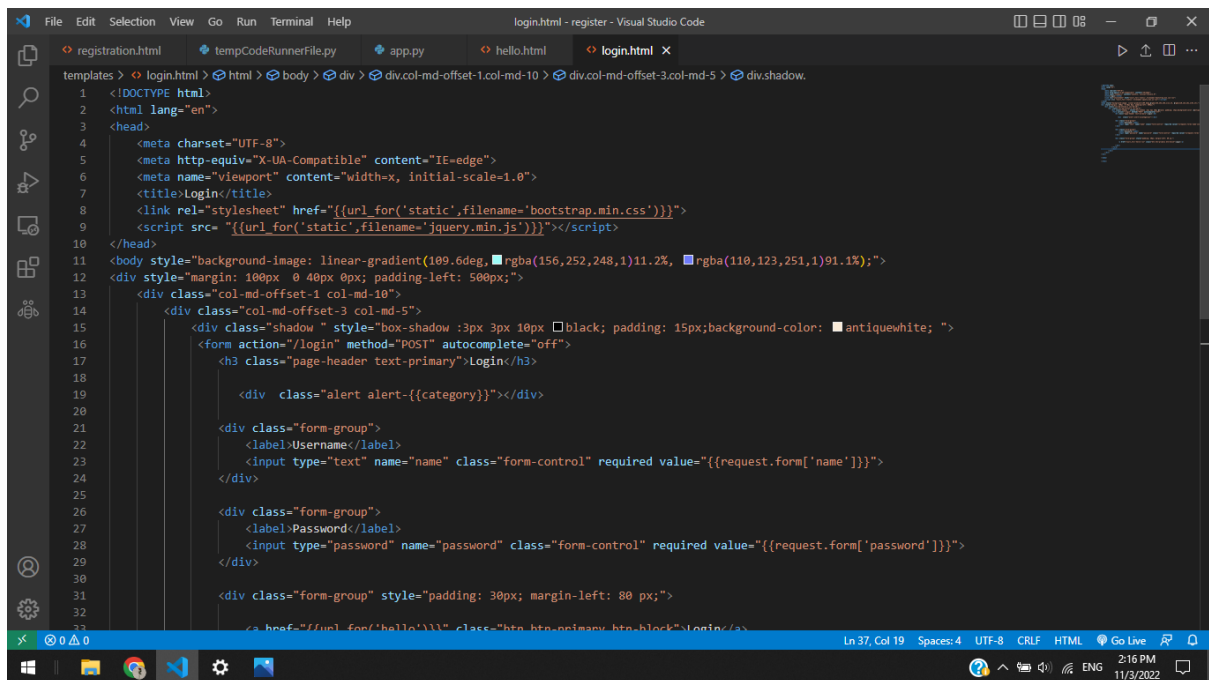
1 from flask import Flask, render_template, request, flash, redirect, url_for, session
2 import ibm_db
3 dsn_hostname="54a2f15b-5c0f-45df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud"
4 dsn_uid="nlf21842"
5 dsn_pwd="yoaheuG1UHSqexn"
6 dsn_driver="IBMDB2CL1"
7 dsn_database="bludb"
8 dsn_port="32733"
9 dsn_protocol="TCP/IP"
10 dsn_security="SSL"
11 dsn_cert="DigiCertGlobalRootCA.crt"
12 dsn={
13     "DRIVER={0};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
14     "DATABASE={1};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
15     "HOSTNAME={2};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
16     "PORT={3};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
17     "PROTOCOL={4};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
18     "UID={5};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
19     "PWD={6};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
20     "SECURITY={7};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
21     "SSLServerCertificate={8};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
22 }
23 conn=ibm_db.connect(dsn, "", "")
24 app=Flask(__name__)
25 @app.route('/')
26 def index():
27     return render_template("registration.html")
28 @app.route('/registration', methods=['GET', 'POST'])
29 def registration():
30     if request.method == 'POST':
31         NAME=request.form['name']
32         EMAIL=request.form['mail']
33         CONTACT=request.form['contact']
34         PASSWORD=request.form['password']

```

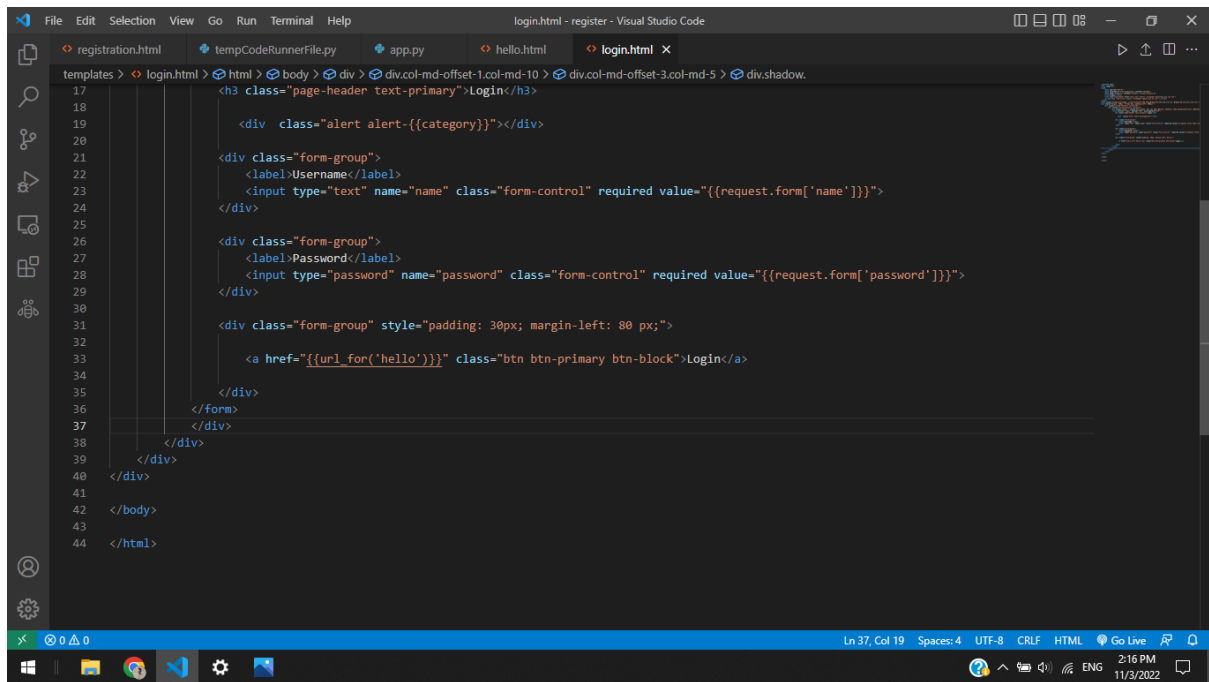


```
26 return render_template("registration.html")
27 @app.route('/registration', methods=['GET', 'POST'])
28 def registration():
29     if request.method == 'POST':
30         NAME=request.form['name']
31         EMAIL=request.form['mail']
32         CONTACT=request.form['contact']
33         PASSWORD=request.form['password']
34         insertQuery="INSERT INTO NUTRITABLE VALUES('"+NAME+"','"+EMAIL+"','"+CONTACT+"','"+PASSWORD+"')"
35         insertStmt=ibm_db.prepare(conn, insertQuery)
36         ibm_db.execute(insertStmt)
37         return redirect(url_for("login"))
38     return render_template("registration.html")
39 @app.route('/login', methods=['GET', 'POST'])
40 def login():
41     if request.method=='POST':
42         name=request.form['name']
43         password=request.form['password']
44         query="select COUNT(*) from nutritible where name='"+name+"' and password='"+password+"'"
45         stmt5=ibm_db.exec_immediate(conn,query)
46         row=ibm_db.fetch_tuple(stmt5)
47         if(row[0]==1):
48             return redirect(url_for("hello"))
49     return render_template("login.html")
50 @app.route('/hello', methods=['GET', 'POST'])
51 def hello():
52     return render_template("hello.html")
53 if __name__ == "__main__":
54     app.run(debug=True)
```

Login.html:

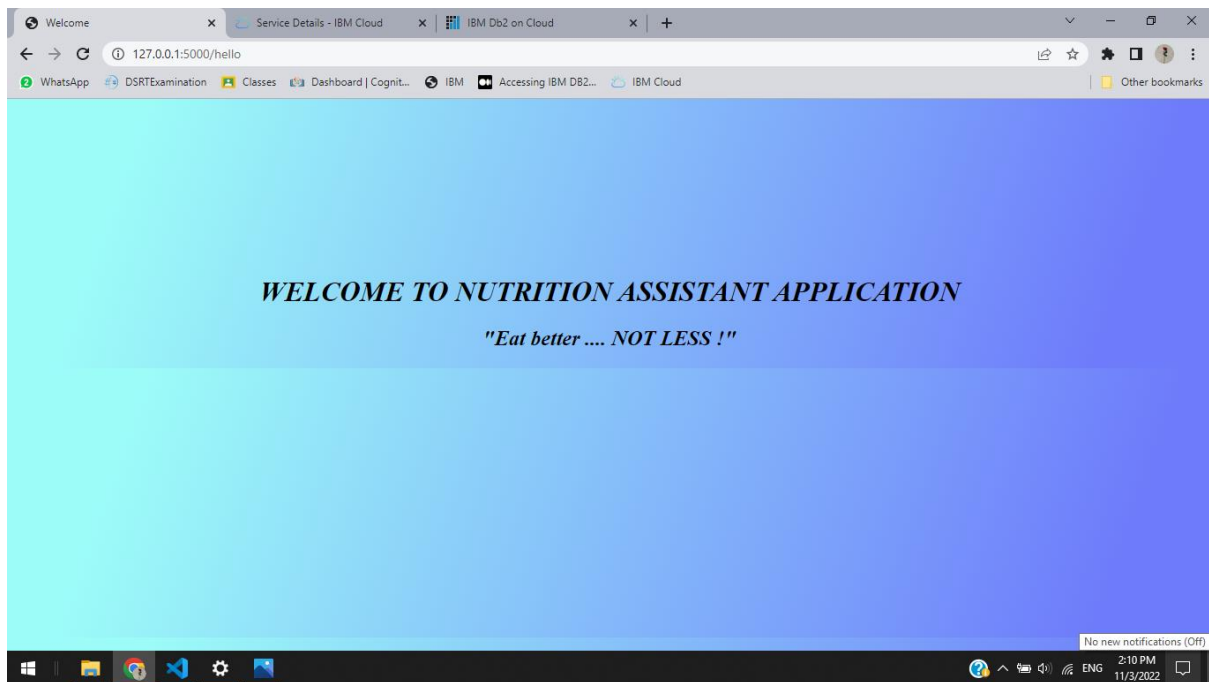


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>login</title>
8     <link rel="stylesheet" href="{{url_for('static', filename='bootstrap.min.css')}}">
9     <script src="{{url_for('static', filename='jquery.min.js')}}"></script>
10 </head>
11 <body style="background-image: linear-gradient(109.6deg, #1565c0 1%, #1976d2 1%, #1e88e5 1%);">
12 <div style="margin: 100px 0 40px 0px; padding-left: 500px;">
13     <div class="col-md-offset-1 col-md-5">
14         <div class="col-md-offset-3 col-md-5">
15             <div class="shadow" style="box-shadow: 3px 3px 10px #000; padding: 15px; background-color: #f9f9f9;">
16                 <form action="/login" method="POST" autocomplete="off">
17                     <h3 class="page-header text-primary">Login</h3>
18
19                     <div class="alert alert-{{category}}"></div>
20
21                     <div class="form-group">
22                         <label>Username</label>
23                         <input type="text" name="name" class="form-control" required value="{{request.form['name']}}">
24                     </div>
25
26                     <div class="form-group">
27                         <label>Password</label>
28                         <input type="password" name="password" class="form-control" required value="{{request.form['password']}}">
29                     </div>
30
31                     <div class="form-group" style="padding: 30px; margin-left: 80px;">
32                         <button type="submit" value="Login" class="btn btn-primary btn-block">Login</button>
33                     </div>
34                 </form>
35             </div>
36         </div>
37     </div>
38 </body>
```

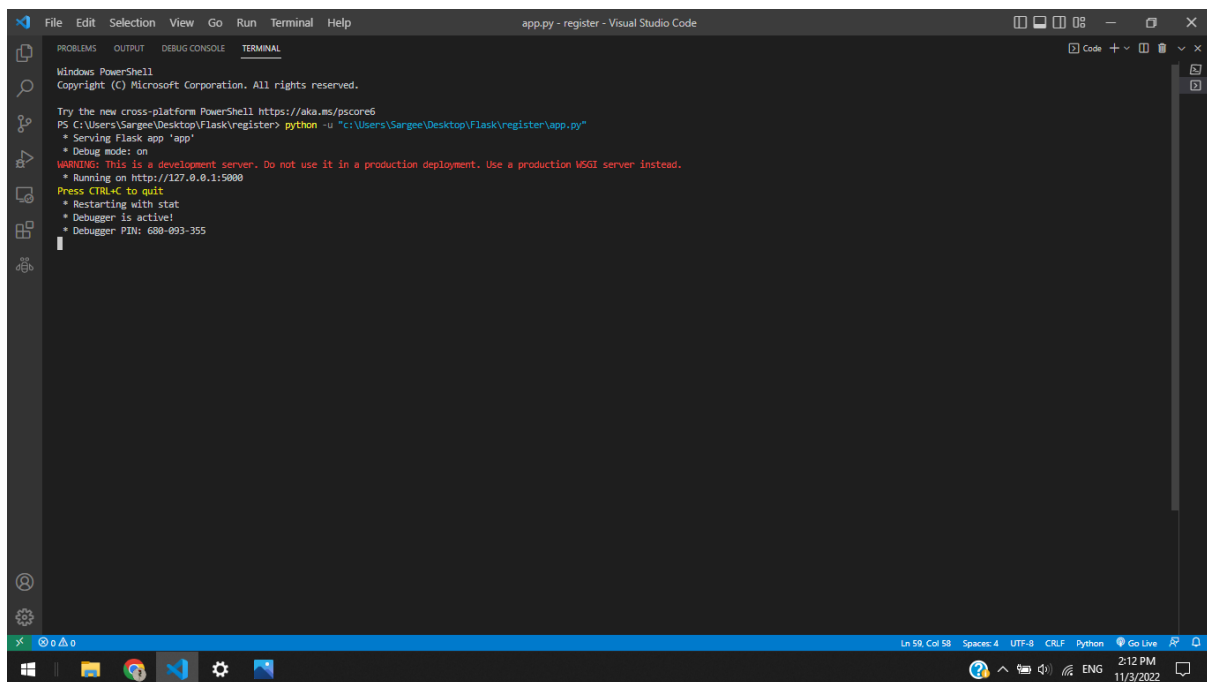


```
17 <h3 class="page-header text-primary">Login</h3>
18
19 <div class="alert alert-{{category}}"></div>
20
21 <div class="form-group">
22 <label>Username</label>
23 <input type="text" name="name" class="form-control" required value="{{request.form['name']}}">
24 </div>
25
26 <div class="form-group">
27 <label>Password</label>
28 <input type="password" name="password" class="form-control" required value="{{request.form['password']}}">
29 </div>
30
31 <div class="form-group" style="padding: 30px; margin-left: 80 px;">
32
33 <a href="{{url_for('hello')}}" class="btn btn-primary btn-block">Login</a>
34
35 </div>
36 </form>
37 </div>
38 </div>
39 </div>
40 </div>
41
42 </body>
43
44 </html>
```

WelcomePage.html :



Terminal :



```
File Edit Selection View Go Run Terminal Help
app.py - register - Visual Studio Code

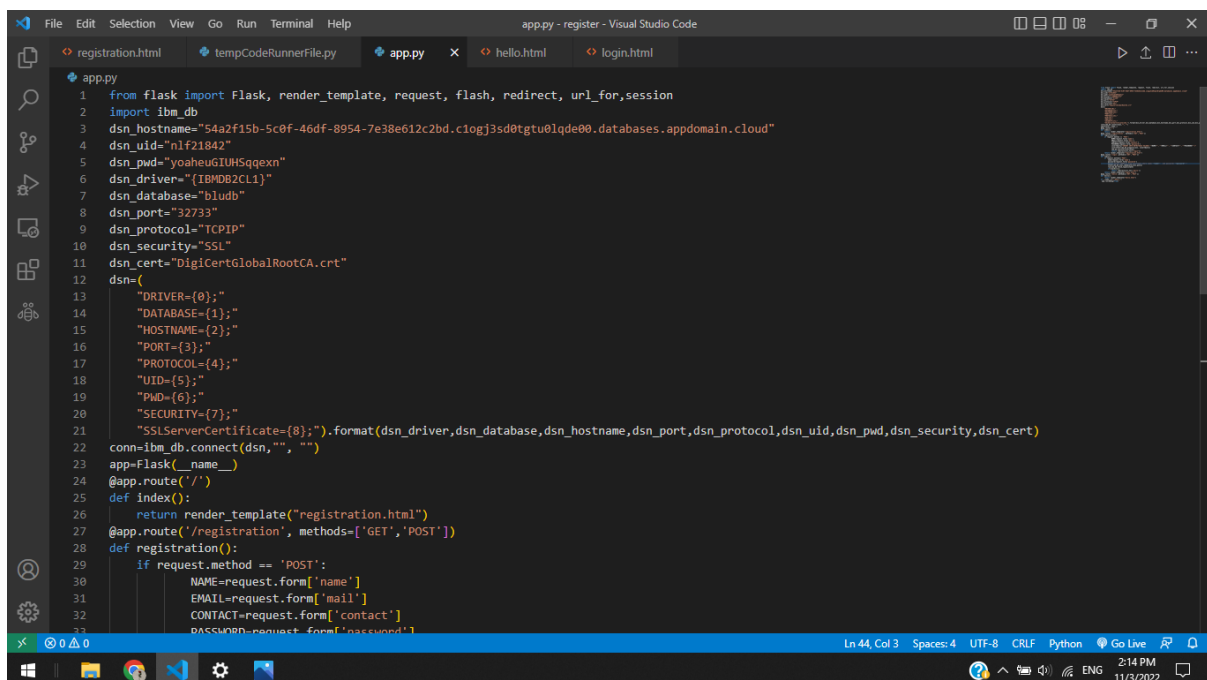
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell
PS C:\Users\Sargee\Desktop\Flask\register> python -u "c:\Users\Sargee\Desktop\Flask\register\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 688-893-355

Ln 59, Col 58 Spaces: 4 UTF-8 CRLF Python Go Live
```

App.py :



```
File Edit Selection View Go Run Terminal Help
app.py - register - Visual Studio Code

registration.html tempCodeRunnerFile.py app.py x hello.html login.html

app.py
1 from flask import Flask, render_template, request, flash, redirect, url_for, session
2 import ibm_db
3 dsn_hostname="54a2f15b-5c0f-46df-8954-7e38e612c2bd.c10g13sd0tgtu0lqde00.databases.appdomain.cloud"
4 dsn_uid="nlf21842"
5 dsn_pwd="yohaheuGIUHSqgexn"
6 dsn_driver="(IBMDB2CL1)"
7 dsn_database="bludb"
8 dsn_port="32733"
9 dsn_protocol="TCPIP"
10 dsn_security="SSL"
11 dsn_cert="DigiCertGlobalRootCA.crt"
12 dsn={
13     "DRIVER={0};"
14     "DATABASE={1};"
15     "HOSTNAME={2};"
16     "PORT={3};"
17     "PROTOCOL={4};"
18     "UID={5};"
19     "PWD={6};"
20     "SECURITY={7};"
21     "SSLServerCertificate={8};".format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_protocol,dsn_uid,dsn_pwd,dsn_security,dsn_cert)
22 conn=ibm_db.connect(dsn,"", "")
23 app=Flask(__name__)
24 @app.route('/')
25 def index():
26     return render_template("registration.html")
27 @app.route('/registration', methods=['GET','POST'])
28 def registration():
29     if request.method == 'POST':
30         NAME=request.form['name']
31         EMAIL=request.form['mail']
32         CONTACT=request.form['contact']
33         PASSWORD=request.form['password']

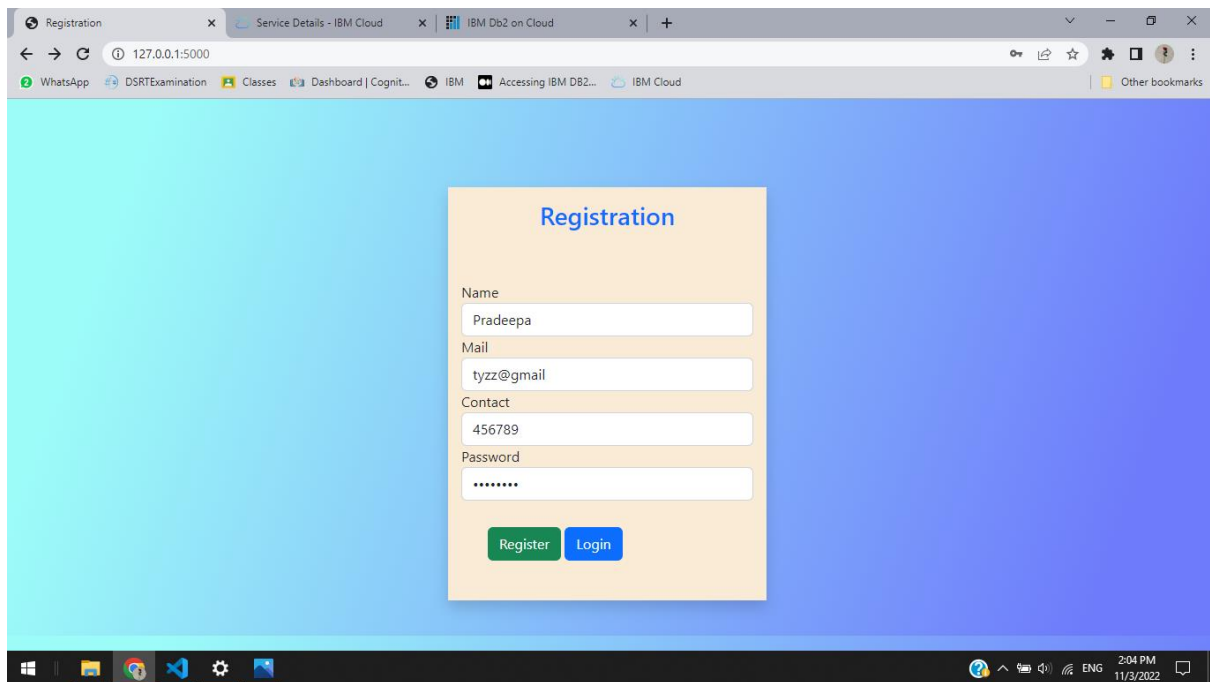
Ln 44, Col 3 Spaces: 4 UTF-8 CRLF Python Go Live
```



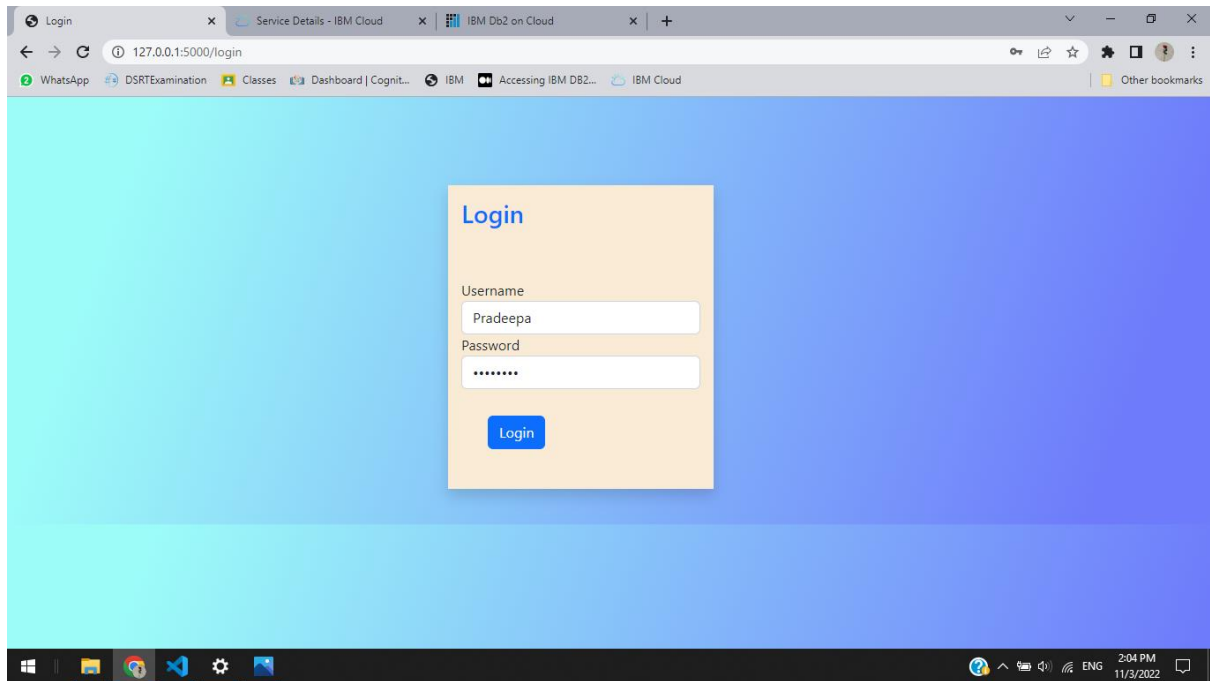
```
File Edit Selection View Go Run Terminal Help
app.py - register - Visual Studio Code
registration.html tempCodeRunnerFile.py app.py x hello.html login.html
app.py
26 return render_template("registration.html")
27 @app.route('/registration', methods=['GET', 'POST'])
28 def registration():
29     if request.method == 'POST':
30         NAME=request.form['name']
31         EMAIL=request.form['mail']
32         CONTACT=request.form['contact']
33         PASSWORD=request.form['password']
34         insertQuery="INSERT INTO NUTRITABLE VALUES('"+NAME+"','"+EMAIL+"','"+CONTACT+"','"+PASSWORD+"')"
35         insertStmt=ibm_db.prepare(conn, insertQuery)
36         ibm_db.execute(insertStmt)
37         return redirect(url_for("login"))
38     return render_template("registration.html")
39 @app.route('/login', methods=['GET', 'POST'])
40 def login():
41     if request.method == 'POST':
42         name=request.form['name']
43         password=request.form['password']
44         query="select COUNT(*) from nutritable where name='"+name+"' and password='"+password+"'"
45         stmt5=ibm_db.exec_immediate(conn, query)
46         row=ibm_db.fetch_tuple(stmt5)
47         if(row[0]==1):
48             return redirect(url_for('hello'))
49     return render_template("login.html")
50 @app.route('/hello', methods=['GET', 'POST'])
51 def hello():
52     return render_template("hello.html")
53 if __name__ == "__main__":
54     app.run(debug=True)
```

OUTPUT :

REGISTERING :



LOGIN :



DATABASE :

