

Assignment -2

Python Flask

Assignment Date	27 September 2022
Student Name	Mr. Prabu K
Student Roll Number	711319CS111
Maximum Marks	2 Marks

Question-1:

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

Solution:

The screenshot displays the IBM Db2 on Cloud web interface. The top navigation bar includes options like 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected. On the left, a sidebar shows a search bar and a list of tables. The main area shows a table definition for 'USERS' in schema 'YBK70810'. The table has four columns: USERNAME, EMAIL, PASSWORD, and ROLLNO, all of type CHAR with a length of 50 and are nullable. A 'View data' button is visible at the bottom right of the table definition panel.

Name	Schema	Properties
USERS	YBK70810	...

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	Y	50	0
EMAIL	CHAR	Y	50	0
PASSWORD	CHAR	Y	50	0
ROLLNO	CHAR	Y	50	0

Question-2:

2. Perform UPDATE,DELETE Queries with user table?

Solution:

Update

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

YBK70810

Tables

USERS

Views

MQTs

Aliases

Nicknames

*Untitled - 3

1 UPDATE "YBK70810"."USERS" set rollno='19cs101' where username='TLF3411';

2

Syntax assistant

Run all

History

Find history

Script	Date	Status	Runtime
Untitled - 3	Nov 2, 2022 12:57:15 PM	✓ 1	0.005 s
UPDATE "YBK70810"."USERS" set rollno='19cs101' where username='...		✓	0.005 s

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/.../index.html

Delete

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

YBK70810

Tables

USERS

Views

MQTs

Aliases

Nicknames

*Untitled - 3

1 DELETE FROM "YBK70810"."USERS" where username='Naveen';

2

Syntax assistant

Run all

History

Find history

Script	Date	Status	Runtime
Untitled - 3	Nov 2, 2022 12:54:28 PM	✓ 1	0.006 s
DELETE FROM "YBK70810"."USERS" where username='Naveen'		✓	0.006 s

Question-3:

3. Connect python code to db2?

Solution:

```

connection.py x
connection.py > ...
1 import ibm_db
2
3 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-0fb7e483086.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32459;SECURITY=SSL;SSLServerCertificate=D
4
5 print(conn)

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** Python

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS D:\IBM\Our Work\Preparation Phase\Assignment 2> & C:/Users/naveen/AppData/Local/Programs/Python/Python37/python.exe "d:/IBM/Our Work/Preparation Phase/Assignment 2/connection.py"

<ibm_db.IBM_DBConnection object at 0x000025E47C82930>

PS D:\IBM\Our Work\Preparation Phase\Assignment 2>

Question-4:

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?

Solution:

Display.html

```
display.html X
templates > display.html > 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      <link rel="stylesheet" href="../static/CSS/display.css" />
8      <title>Welcome Page</title>
9  </head>
10 <body style="background-color: #blanchedalmond;">
11
12     <div class="card">
13         <h1>Welcome</h1>
14         <h2>{{name}}</h2>
15         <p class="title">{{ msg }}</p>
16         <p></p>
17
18         <p><button>IBM</button></p>
19     </div>
20
21 </body>
22 </html>
23
```

Login.html

login.html X

templates > login.html > 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 Page</title>
8      <link rel="stylesheet" href="../static/CSS/style.css" />
9  </head>
10 <body>
11     <div class="container">
12         <div class="login text-center mt-5">
13             <h1>Login Form</h1>
14             <hr>
15             <form action="/login" method="post">
16                 <div class="msg">{{ msg }}</div>
17                 <label for="username"><b>Username:</b></label>
18                 <input type="text" name="username" placeholder="Enter Your Username" id="username" required />
19                 <label for="password"><b>Password:</b></label>
20                 <input type="password" name="password" placeholder="Enter Your Password" id="password" required />
21                 <button type="submit" id="button" class="registerbtn">Login</button>
22             </form>
23         </div>
24         <div class="note mt-3 text-center">
25             <p>
26                 Don't have an account yet? Click here to <a href="/">register! </a>
27             </p>
28         </div>
29     </div>
30 </body>
31 </html>
```

Register.html

register.html X

templates > register.html > html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
5    <link rel="stylesheet" href="../../static/CSS/style.css" />
6    <title>Registration page</title>
7  </head>
8  <body>
9    <form action="http://localhost:3030/" method="post">
10     <div class="container">
11       <h1>Registration page</h1><hr />
12       <div class="msg">{{ msg }}</div>
13       <label for="username"><b>Username:</b></label>
14       <input type="text" placeholder="Enter your Username" name="username" id="username" required />
15       <label for="email"><b>Email:</b></label>
16       <input type="text" placeholder="Enter your Email" name="email" id="email" required />
17       <label for="rollno"><b>Roll number:</b></label>
18       <input type="text" id="rollno" name="rollno" placeholder="Enter your Rollno" />
19       <label for="password"><b>Password:</b></label>
20       <input type="password" id="password" name="password" placeholder="Enter your password" />
21       <button type="submit" class="registerbtn">Register</button>
22       <div class="note mt-3 text-center">
23         <p> already have an account ? please login <a href="/login">login! </a> </p>
24       </div>
25     </div>
26   </form>
27 </body>
28 </html>
```

Display.css

```
static > CSS > # display.css > .card
1  .card {
2      box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
3      max-width: 300px;
4      margin: auto;
5      text-align: center;
6      font-family: arial;
7      padding-top: 10px;
8      margin-top: 50px;
9      background-color: white;
10 }
11 .title {
12     color: grey;
13     font-size: 18px;
14 }
15 button {
16     border: none;
17     outline: 0;
18     display: inline-block;
19     padding: 8px;
20     color: white;
21     background-color: #000;
22     text-align: center;
23     cursor: pointer;
24     width: 100%;
25     font-size: 18px;
26 }
27 a {
28     text-decoration: none;
29     font-size: 22px;
30     color: black;
31 }
32 button:hover, a:hover {
33     opacity: 0.7;
34 }
```

Style.css

```
# style.css X
static > CSS > # style.css > ⚙️ .registerbtn:hover
1  body {
2      font-family: Arial, Helvetica, sans-serif;
3      background-color: white
4  }
5  * {
6      box-sizing: border-box;
7  }
8  .container {
9      padding: 100px;
10     background-color: white;
11     margin: 10px;
12 }
13 }
14 h1{
15     text-align: center;
16 }
17 a{
18     text-decoration: none;
19     color: blue;
20 }
21 input {
22     width: 100%;
23     padding: 15px;
24     margin: 5px 0 22px 0;
25     display: inline-block;
26     border: none;
27     background: #f1f1f1;
28 }
29 .registerbtn {
30     background-color: blue;
31     color: white;
32     padding: 16px 20px;
33     margin: 8px 0;
34     border: none;
35     cursor: pointer;
36     opacity: 0.9;
37 }
38 .registerbtn:hover {
39     opacity: 1;
40 }
```

App.py

```

app.py > ...
1 from flask import Flask, render_template, request, redirect, url_for, session
2 import ibm_db
3 import re
4
5 app = Flask(__name__)
6 app.secret_key = 'a'
7 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-0fb7e483086.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32459;SECURITY=SSL;SSLServerCertificate=D
8
9 @app.route('/', methods = ['GET', 'POST'])
10 def register():
11     msg = ''
12     if request.method == 'POST' :
13         username = request.form['username']
14         email = request.form['email']
15         rollno = request.form['rollno']
16         password = request.form['password']
17         sql = "SELECT * FROM users WHERE username=?"
18         stmt = ibm_db.prepare(conn, sql)
19         ibm_db.bind_param(stmt,1,username)
20         ibm_db.execute(stmt)
21         account = ibm_db.fetch_assoc(stmt)
22         print(account)
23         if account:
24             msg = 'Account already exists !'
25         elif not re.match(r'[^@]+\.[^@]+\.[^@]+', email):
26             msg = 'Invalid email address !'
27         elif not re.match(r'[A-Za-z0-9]+', username):
28             msg = 'name must contain only characters and numbers !'
29         else:
30             insert_sql = "INSERT INTO users VALUES (?, ?, ?, ?)"
31             prep_stmt = ibm_db.prepare(conn, insert_sql)
32             ibm_db.bind_param(prepare_stmt, 1, username)
33             ibm_db.bind_param(prepare_stmt, 2, email)
34             ibm_db.bind_param(prepare_stmt, 3, password)
35             ibm_db.bind_param(prepare_stmt, 4, rollno)
36             ibm_db.execute(prepare_stmt)
37             msg = 'You have successfully registered !'
38         elif request.method == 'POST':
39             msg = 'Please fill out the form !'
40     return render_template('register.html', msg = msg)

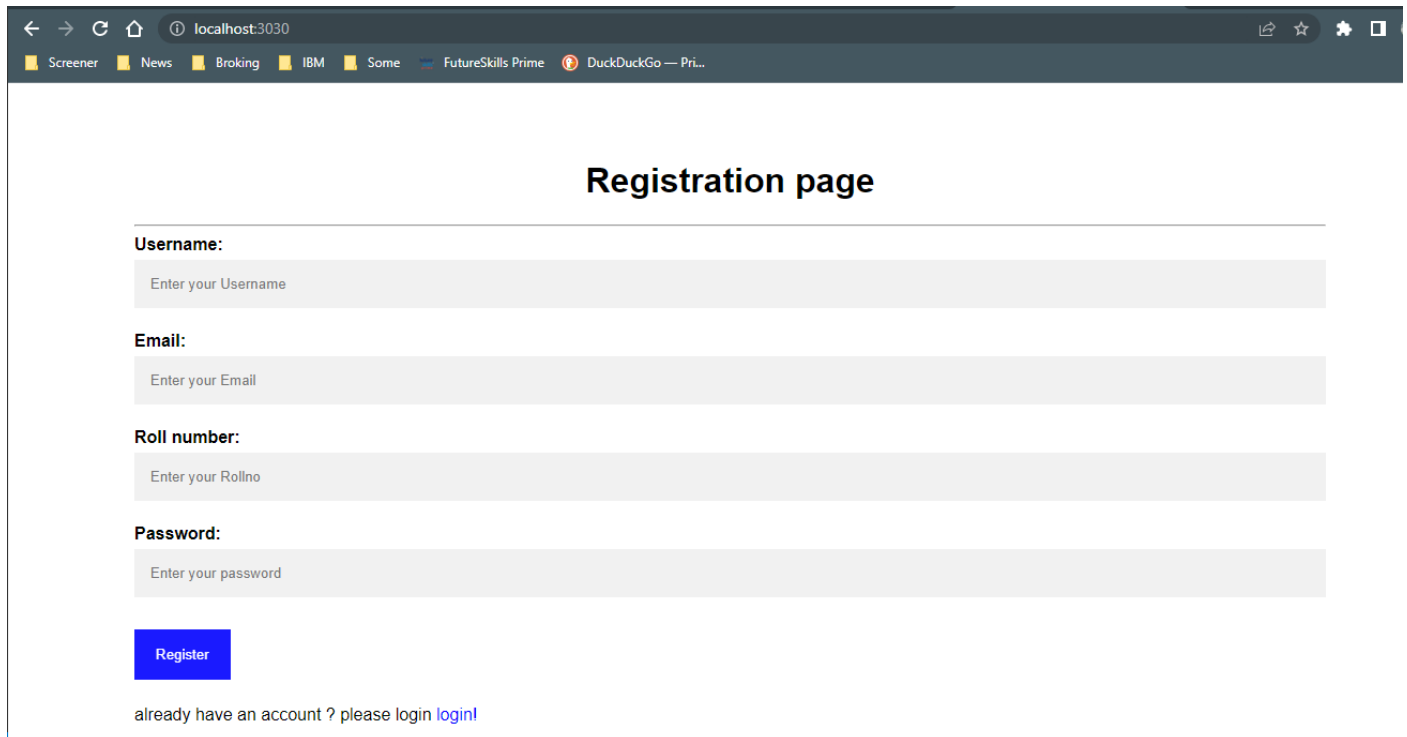
```

```

app.py X
app.py > ...
35         ibm_db.bind_param(prepare_stmt, 4, rollno)
36         ibm_db.execute(prepare_stmt)
37         msg = 'You have successfully registered !'
38     elif request.method == 'POST':
39         msg = 'Please fill out the form !'
40     return render_template('register.html', msg = msg)
41
42 @app.route('/login', methods = ['GET', 'POST'])
43 def login():
44     global userid
45     msg = ''
46
47     if request.method == 'POST' :
48         username = request.form['username']
49         password = request.form['password']
50         sql = "SELECT * FROM Users WHERE username=? AND password=?"
51         stmt = ibm_db.prepare(conn, sql)
52         ibm_db.bind_param(stmt,1,username)
53         ibm_db.bind_param(stmt,2,password)
54         ibm_db.execute(stmt)
55         account = ibm_db.fetch_assoc(stmt)
56         print (account)
57         if account:
58             session['loggedin'] = True
59             session['id'] = account['USERNAME']
60             userid= account['USERNAME']
61             session['username'] = account['USERNAME']
62             msg = 'Logged in successfully !'
63             return render_template('display.html', msg = msg, name = account['USERNAME'])
64         else:
65             msg = 'Incorrect username / password !'
66     return render_template('login.html', msg = msg)
67
68 if __name__ == '__main__':
69     app.run(["localhost",3030,True])

```


Registration Page



Registration page

Username:
Enter your Username

Email:
Enter your Email

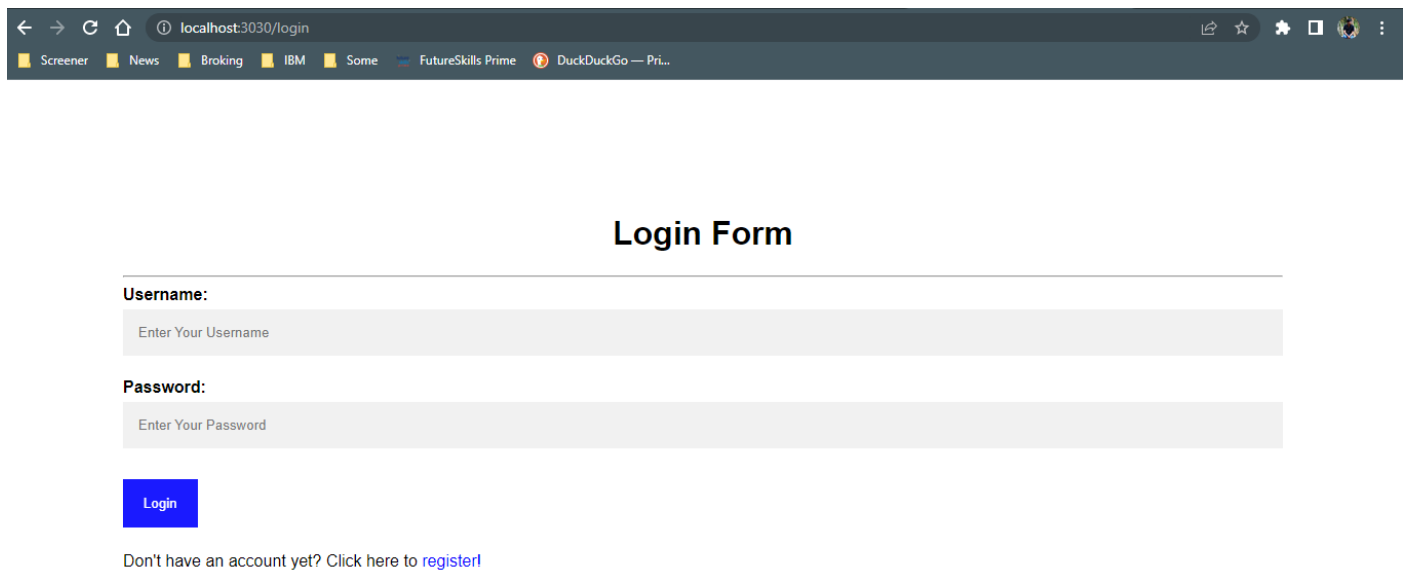
Roll number:
Enter your Rollno

Password:
Enter your password

Register

already have an account ? please login [login!](#)

Login Page



Login Form

Username:
Enter Your Username

Password:
Enter Your Password

Login

Don't have an account yet? Click here to [register!](#)

Welcome Page

Welcome

TLF3411

Logged in successfully !

IBM