

## Assignment-2

Assignment Date	29 September 2022
Student Name	Snehalatha P
Student Roll Number	511319205029
Maximum Marks	2 Marks

### Questions:

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

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, and the 'USER' table is highlighted. The 'Table definition' panel on the right shows the table structure:

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	Y	20	0
ROLLNUMBER	INTEGER	Y		0
EMAIL	CHAR	Y	25	0
PASSWORD	VARCHAR	Y	32	0

2. Perform Update, Delete Queries with user table.

### Update Table:

The screenshot shows the IBM Db2 on Cloud web interface with a SQL query executed. The query is:

```
INSERT INTO USER VALUES('Mayurisri','14','mayurisriam412@gmail.com','93601')
```

The 'History' panel shows the execution details:

Script	Date	Status	Runtime
Untitled - 1	Nov 5, 2022 10:01:11 PM	✓ 1	0.009 s
INSERT INTO USER VALUES('Mayurisri','14','mayurisriam412@gmail.com','93601')		✓	0.009 s

Service Details - IBM Cloud x IBM Db2 on Cloud x +

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F0725f098320b4d748cea66e1745cc408%3A0ae6eb0f-6cb...

IBM Db2 on Cloud

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

DMD92141.USER Back

Export to CSV

USERNAME	ROLLNUMBER	EMAIL	PASSWORD
Mayurisri	14	mayurisriram412@gmail.com	93601
kavini	11	kavini11@gmail.com	110418

21°C Partly cloudy ENG IN 22:06 05-11-2022

## Delete Table:

Service Details - IBM Cloud x IBM Db2 on Cloud x +

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F0725f098320b4d748cea66e1745cc408%3A0ae6eb0f-6cb...

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

DMD92141

\*Untitled - 1 x +

1 DELETE FROM USER WHERE USERNAME='Mayurisri'

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 5, 2022 10:09:41 PM	✓ 1	0.010 s
DELETE FROM USER WHERE USERNAME='Mayurisri'		✓	0.010 s
Untitled - 1	Nov 5, 2022 10:04:25 PM	✓ 1	0.008 s
INSERT INTO USER VALUES('kavini','11','kavini11@gmail.com','110418')		✓	0.008 s
Untitled - 1	Nov 5, 2022 10:01:11 PM	✓ 1	0.009 s
INSERT INTO USER VALUES('Mayurisri','14','mayurisriram412@gmail.com','93601')		✓	0.009 s

21°C Partly cloudy ENG IN 22:11 05-11-2022

Service Details - IBM Cloud | IBM Db2 on Cloud

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

DMD92141.USER [Back]

Export to CSV

USERNAME	ROLLNUMBER	EMAIL	PASSWORD
kavini	11	kavini11@gmail.com	110418

21°C Partly cloudy | ENG IN | 22:11 05-11-2022

3. Connect python code to db2.

**Note: Question 4 contains Question 3 answer**

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 loginpage authenticate user username and password. If the user is valid show the welcome page.

**Answers:**

**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.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;SE
```

```
CURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dmd92141;PWD=XFKsMMXiZXQ9dmWb",",")
```

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

```
def register():
```

```
    msg = "
```

```
    if request.method == 'POST':
```

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

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

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

```
        sql = "SELECT * FROM user WHERE username =?"
```

```
        stmt = ibm_db.prepare(conn, sql)
```

```
        ibm_db.bind_param(stmt, 1, username)
```

```
        ibm_db.execute(stmt)
```

```
        account = ibm_db.fetch_assoc(stmt)
```

```
        print(account)
```

```
        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 = 'name must contain only characters and numbers !'
```

```
        else:
```

```
            insert_sql = "INSERT INTO user VALUES (?, ?, ?)"
```

```
            prep_stmt = ibm_db.prepare(conn, insert_sql)
```

```
            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)

@app.route('/login', methods=['GET', 'POST'])
def login():
    global userid
    msg = ''

    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM user WHERE username =? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['loggedin'] = True
            session['id'] = account['USERNAME']
            userid = account['USERNAME']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'

        msg = 'Logged in successfully !'
        return render_template('dashboard.html', msg=msg)

```

else:

msg = 'Incorrect username / password !'

return render\_template('login.html', msg=msg)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0')

# app.run(debug=True)

### **Register.html:**

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="\static\style.css">

</head>

<body>

<form action="/" method="post">

<div class="container">

<center> <h1> Registration Form</h1> </center>

<hr>

<input type="text" name="username" placeholder="Enter username" size="15" required />

<input type="text" name="email" placeholder="Enter email" required>

<input type="password" name="password" placeholder="Enter Password" size="10" / required>

<button type="submit" class="registerbtn">Register</button>

<div class="note mt-3 text-center">

<!--Register form -->

<p> already have an account ? please login <a href="/login">login! </a> </p>

</form>

</body>

</html>

### **Login.html:**

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="\static\style.css">

</head>

<body>

<form method="post" action="login">

<div class="container">

<center> <h1> Login Form</h1> </center>

<hr>

<label><b>UserName:</b></label>

<input type="text" name="username" placeholder="Username" size="15"  
required />

<label><b>Password:</b>

</label>

<input type="password" name="password" placeholder="Password" size="10"/  
required>

<button type="submit" class="registerbtn">Login</button>

<div class="note mt-3 text-center">

<!--Register form -->

<p> Don't have an account yet? Click here to <a href="/">register! </a>

</p>

```
        </div>
    </form>
</body>
</html>
```

### **style.css:**

```
body{
    font-family: Calibri, Helvetica, sans-serif;
    background-color: pink;
}
.container {
    padding: 30px;
    background-color: lightblue;
}

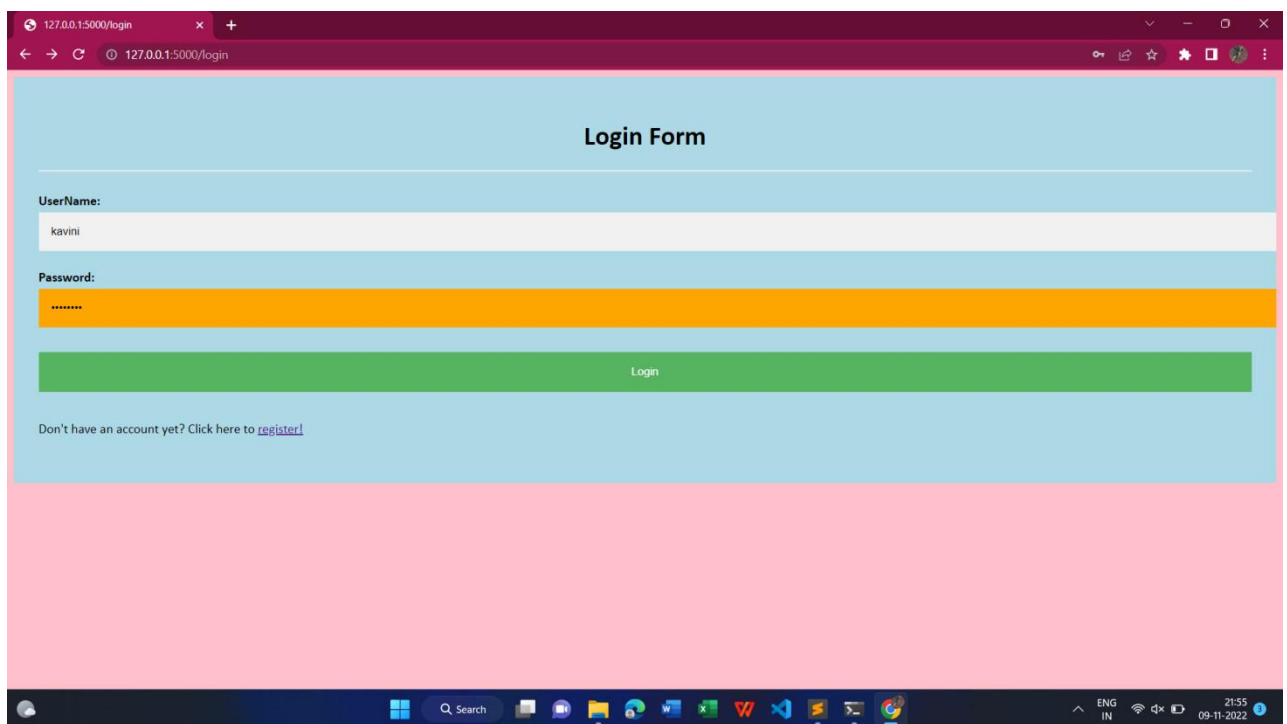
input[type=text], input[type=password], textarea {
    width: 100%;
    padding: 15px;
    margin: 5px 0 22px 0;
    display: inline-block;
    border: none;
    background: #f1f1f1;
}
input[type=text]:focus, input[type=password]:focus {
    background-color: orange;
    outline: none;
}
div {
    padding: 10px 0;
}
hr {
    border: 1px solid #f1f1f1;
    margin-bottom: 25px;
}
.registerbtn {
    background-color: #4CAF50;
    color: white;
    padding: 16px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;
```



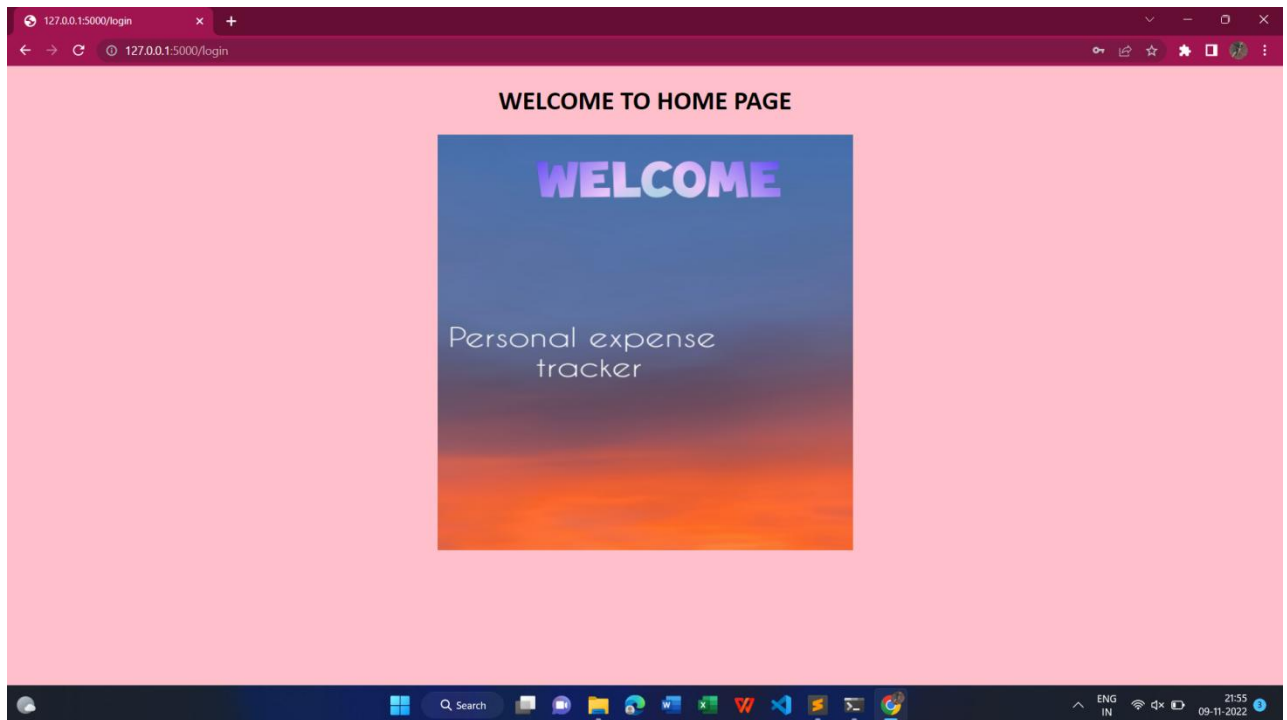
```
width: 100%;  
opacity: 0.9;  
}  
.registerbtn:hover {  
  opacity: 1;  
}
```

**Output:**

### Registration Page:



## Welcome Page:



## Data Base Table:

