

Team ID	PNT2022PMID49593
Project Name Project	PLASMA DONOR APPLICATION
Name	NANTHINI.S
Roll Number	950019104031

Assignment-2

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

```

MySQL 8.0 Command Line Client
mysql> insert into user values('shri@gmail.com','shri',42,100);
Query OK, 1 row affected (0.02 sec)

mysql> select * from user;
+-----+-----+-----+-----+
| email          | username | rollnumber | password |
+-----+-----+-----+-----+
| sindhu@gmail.com | sindhu   | 43         | 0         |
| nanthini10@gmail.com | nanthini | 31         | 123      |
| shri@gmail.com   | shri    | 42         | 100      |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> desc user;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| email | varchar(30) | YES | | NULL | |
| username | varchar(15) | YES | | NULL | |
| rollnumber | int | YES | | NULL | |
| password | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

2.Perform UPDATE, DELETE Queries with user table.

```
MySQL 8.0 Command Line Client

mysql> select * from user;
+-----+-----+-----+-----+
| email          | username | rollnumber | password |
+-----+-----+-----+-----+
| sindhu@gmail.com | sindhu   | 43         | 0         |
| nanthini10@gmail.com | nanthini | 31         | 123       |
| shri@gmail.com   | shrinathi | 42         | 100       |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> delete from user where username='sindhu';
Query OK, 1 row affected (0.02 sec)

mysql> select * from user;
+-----+-----+-----+-----+
| email          | username | rollnumber | password |
+-----+-----+-----+-----+
| nanthini10@gmail.com | nanthini | 31         | 123       |
| shri@gmail.com   | shrinathi | 42         | 100       |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client

+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| email | varchar(30) | YES | | NULL | |
| username | varchar(15) | YES | | NULL | |
| rollnumber | int | YES | | NULL | |
| password | int | YES | | NULL | |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> update user set usernames='shrinathi' where rollnumber=42;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from user;
+-----+-----+-----+-----+
| email          | username | rollnumber | password |
+-----+-----+-----+-----+
| sindhu@gmail.com | sindhu   | 43         | 0         |
| nanthini10@gmail.com | nanthini | 31         | 123       |
| shri@gmail.com   | shrinathi | 42         | 100       |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

3.Connect python code to db2.

App.py

```

from flask import Flask, render_template ,request,redirect,session,url_for

import ibm_db

import re

app = Flask(__name__)

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=mpf67194;PWD=iCG3PvOYv DiHMrq8","")

```

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.

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=125f9f61-9715-46f9-9399-
c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30426;SECURITY=SSL;SSLServe
rCertificate=DigiCertGlobalRootCA.crt;UID=gdy06262;PWD=vWZczlsu0eXyvMld","")

@app.route('/')

def home():

    return render_template('register.html')

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

def login():

    global userid

```

```

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':

        username = request.form['username']

        password = request.form['password']

        email = request.form['email']

        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 = 'Username must contain only characters and numbers !'

        elif not username or not password or not email:

            msg = 'Please fill out the form !'

        else:

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

            stmt = ibm_db.prepare(conn,insert_sql)

            ibm_db.bind_param(stmt, 1, username)

```

```

        ibm_db.bind_param(stmt, 2, password)

        ibm_db.bind_param(stmt, 3, email)

        ibm_db.execute(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.run(debug = True)

```

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>

    <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>
```

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>

```

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>

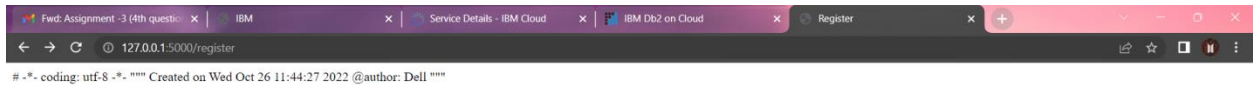
    </div>

  </div>

</body>

</html>
```

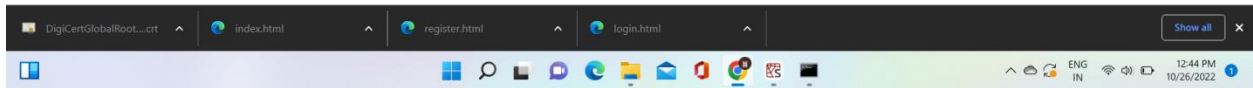
OUTPUT



Register

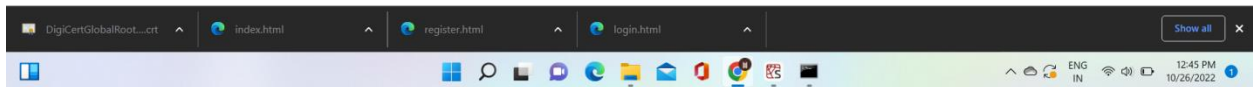
You have successfully registered !

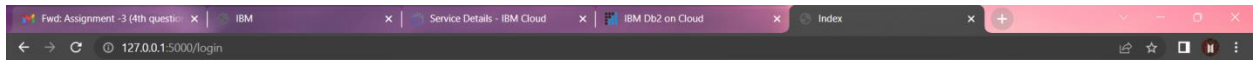
Already have an account? [Sign In here](#)



Login

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



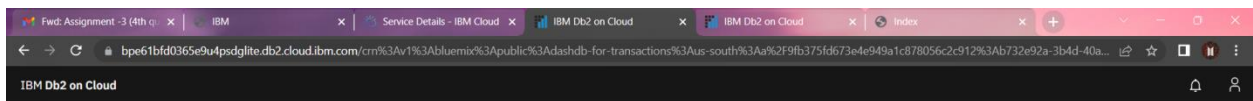
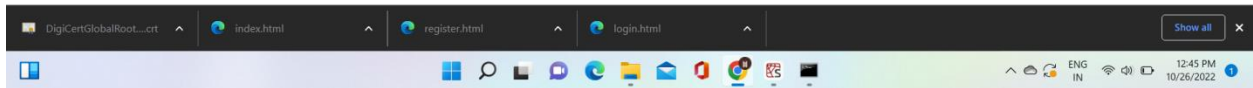


Index

Hi nanthini !!

Welcome to the index page...

[Logout](#)



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

GDY06262.USER Back

Export to CSV

USERNAME	PASSWORD	EMAIL
nanthini	789	950019104035@smartinternz.com
shri	shri	950019104031@smartinternz.com