

ASSIGNMENT 2

Date	12 October 2022
Team ID	PNT2022TMID31636
Maximum Marks	2 Marks

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

Create user table:

The screenshot shows a SQL IDE interface with a script editor and a history panel. The script editor contains the following SQL code:

```
1 CREATE TABLE user_details(  
2     s_no INT GENERATED BY DEFAULT AS IDENTITY NOT NULL,  
3     user_name VARCHAR(150) NOT NULL,  
4     user_email VARCHAR(255) NOT NULL,  
5     roll_no INT NOT NULL,  
6     password VARCHAR(100),  
7     PRIMARY KEY(s_no)  
8 );  
9  
10 insert into user_details(user_name,user_email,roll_no,password) values('Sasi','sasi310@gmail.com',01,'person@123');  
11 insert into user_details(user_name,user_email,roll_no,password) values('Tharini','tharini25@gmail.com',02,'person@123');  
12 insert into user_details(user_name,user_email,roll_no,password) values('Niranjan','niranjan310@gmail.com',03,'person@123');  
13
```

The history panel shows the execution of the script with the following details:

Script	Date	Status	Runtime
Untitled - 2	Nov 14, 2022 11:39:15 AM	3 Success, 1 Error	0.027 s
CREATE TABLE user_details(s_no INT GENERATED BY DEFAULT AS IDENTITY NOT NULL, u...		Error	0.014 s
insert into user_details(user_name,user_email,roll_no,password) values('Sasi','s...		Success	0.004 s
insert into user_details(user_name,user_email,roll_no,password) values('Tharini'...		Success	0.004 s

INSERT DETAILS

* Assignmen... X

* Untitled - 2

+

📄

↶

↷

</>

⌂

🔍

🟢 Syntax assistant

📄

⚙️

Run all

▶

⌵

7 PRIMARY KEY(s_no)

8);

9

10 insert into user_details(user_name,user_email,roll_no,password) values('Sasi','sasi310@gmail.com',01,'person@123');

11 insert into user_details(user_name,user_email,roll_no,password) values('Tharini','tharinim25@gmail.com',02,'person@123');

12 insert into user_details(user_name,user_email,roll_no,password) values('Niranjan','niranjan310@gmail.com',03,'person@123');

13

14

15

16

17

18 update user_details set user_name='sasi kumar' where roll_no='01';

19

20

21

22

23

History

🔍 Find history

🗑️

Script	Date	Status	Runtime	
insert into user_details(user_name,user_email,roll_no,password) values('Tharini...		✔️	0.009 s	:
insert into user_details(user_name,user_email,roll_no,password) values('Niranjan...		✔️	0.009 s	:
update user_details set user_name='sasi kumar' where roll_no='01'		✔️	0.009 s	:

USER TABLE

KCV39089.USER_DETAILS					Back
					Export to CSV
S_NO	USER_NAME	USER_EMAIL	ROLL_NO	PASSWORD	
1	Sasi	sasi310@gmail.com	1	person@123	
2	Tharini	tharinim25@gmail.com	2	person@123	
3	Niranjan	niranjan310@gmail.com	3	person@123	
4	Sasi	sasi310@gmail.com	1	person@123	
5	Tharini	tharinim25@gmail.com	2	person@123	
6	Niranjan	niranjan310@gmail.com	3	person@123	
7	Sasi	sasi310@gmail.com	1	person@123	
8	Tharini	tharinim25@gmail.com	2	person@123	
9	Niranjan	niranjan310@gmail.com	3	person@123	

2. Perform UPDATE, DELETE Queries with usertable

Update:

* Assignmen... x * Untitled - 2 +

Run all

```
7 PRIMARY KEY(s_no)
8 );
9
10 insert into user_details(user_name,user_email,roll_no,password) values('Sasi','sasi310@gmail.com',01,'person@123');
11 insert into user_details(user_name,user_email,roll_no,password) values('Tharini','tharinim25@gmail.com',02,'person@123');
12 insert into user_details(user_name,user_email,roll_no,password) values('Niranjan','niranjan310@gmail.com',03,'person@123');
13
14
15
16
17
18 update user_details set user_name='sasi kumar' where roll_no='01';
19
20
21
22
23
```

History



Find history

Script	Date	Status	Runtime
insert into user_details(user_name,user_email,roll_no,password) values('Tharini'...		✓	0.009 s
insert into user_details(user_name,user_email,roll_no,password) values('Niranjan'...		✓	0.009 s
update user_details set user_name='sasi kumar' where roll_no='01'		✓	0.009 s

Updated Table:

KCV39089.USER_DETAILS

Back









						Export to CSV 
S_NO	USER_NAME	USER_EMAIL	ROLL_NO	PASSWORD		
1	sasi kumar	sasi310@gmail.com	1	person@123		
2	Tharini	tharinim25@gmail.com	2	person@123		
3	Niranjan	niranjan310@gmail.com	3	person@123		
4	sasi kumar	sasi310@gmail.com	1	person@123		
5	Tharini	tharinim25@gmail.com	2	person@123		
6	Niranjan	niranjan310@gmail.com	3	person@123		
7	sasi kumar	sasi310@gmail.com	1	person@123		
8	Tharini	tharinim25@gmail.com	2	person@123		
9	Niranjan	niranjan310@gmail.com	3	person@123		





Delete:

* Assignmenten... x

* Untitled - 2



+

Syntax assistant   Run all  

```
7 PRIMARY KEY(s_no)
8 );
9
10 insert into user_details(user_name,user_email,roll_no,password) values('Sasi','sasi310@gmail.com','01','person@123');
11 insert into user_details(user_name,user_email,roll_no,password) values('Tharini','tharinim25@gmail.com','02','person@123');
12 insert into user_details(user_name,user_email,roll_no,password) values('Niranjan','niranjan310@gmail.com','03','person@123');
13
14
15
16
17
18 update user_details set user_name='sasi kumar' where roll_no='01';
19
20 delete from user_details where user_name='Niranjan';
21
22
23
```

History

 Find history 









Script	Date	Status	Runtime	
insert into user_details(user_name,user_email,roll_no,password) values('Tharini'...			0.008 s	
insert into user_details(user_name,user_email,roll_no,password) values('Niranjan'...			0.008 s	
update user_details set user_name='sasi kumar' where roll_no='01'			0.006 s	
delete from user_details where user_name='Niranjan'			0.007 s	

Table after performed delete query

KCV39089.USER_DETAILS

Back

S_NO	USER_NAME	USER_EMAIL	ROLL_NO	PASSWORD
1	sasi kumar	sasi310@gmail.com	1	person@123
2	Tharini	tharinim25@gmail.com	2	person@123
4	sasi kumar	sasi310@gmail.com	1	person@123
5	Tharini	tharinim25@gmail.com	2	person@123
7	sasi kumar	sasi310@gmail.com	1	person@123
8	Tharini	tharinim25@gmail.com	2	person@123
10	sasi kumar	sasi310@gmail.com	1	person@123
11	Tharini	tharinim25@gmail.com	2	person@123
13	sasi kumar	sasi310@gmail.com	1	person@123

3. Connect python code todb2.

```
import ibm_db
import bcrypt
conn=ibm_db.connect(" DATABASE =bludb;HOSTNAME=b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=kcv39089;PWD=pthMo96lY5zuNGGs",",")
```

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 with username and password. If the user is valid show the welcomepage

App.py

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

```
import ibm_db
import bcrypt
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCP;IP;UID=dcl74999;PWD=Px6tqEmXL9AcQigs","")
```

```
# url_for('static', filename='style.css')
```

```
app = Flask(__name__)
app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'
```

```
@app.route("/", methods=['GET'])
def home():
    if 'email' not in session:
        return redirect(url_for('login'))
    return render_template('home.html', name='Home')
```

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

```
def register():
```

```
    if request.method == 'POST':
        email = request.form['email']
        username = request.form['username']
        rollNo = request.form['rollNo']
        password = request.form['password']
```

```
    if not email or not username or not rollNo or not password:
        return render_template('register.html', error='Please fill all fields')
```

```
    hash = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
```

```
    query = "SELECT * FROM USER WHERE email=? OR rollNo=?"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt, 1, email)
    ibm_db.bind_param(stmt, 2, rollNo)
    ibm_db.execute(stmt)
    isUser = ibm_db.fetch_assoc(stmt)
```

```
    if not isUser:
        insert_sql = "INSERT INTO USER(EMAIL, USERNAME, ROLLNO, PASSWORD) VALUES (?, ?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, email)
        ibm_db.bind_param(prepare_stmt, 2, username)
```

```

        ibm_db.bind_param(prepare_stmt, 3, rollNo)
        ibm_db.bind_param(prepare_stmt, 4, hash)
        ibm_db.execute(prepare_stmt)
        return render_template('register.html', success="You can login")
    else:
        return render_template('register.html', error='Invalid Credentials')

return render_template('register.html', name='Home')

@app.route("/login", methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']

        if not email or not password:
            return render_template('login.html', error='Please fill all fields')
        query = "SELECT * FROM USER WHERE email=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        isUser = ibm_db.fetch_assoc(stmt)
        print(isUser, password)

        if not isUser:
            return render_template('login.html', error='Invalid Credentials')

        isPasswordMatch = bcrypt.checkpw(password.encode('utf-8'), isUser['PASSWORD'].encode('utf-8'))

        if not isPasswordMatch:
            return render_template('login.html', error='Invalid Credentials')

        session['email'] = isUser['EMAIL']
        return redirect(url_for('home'))

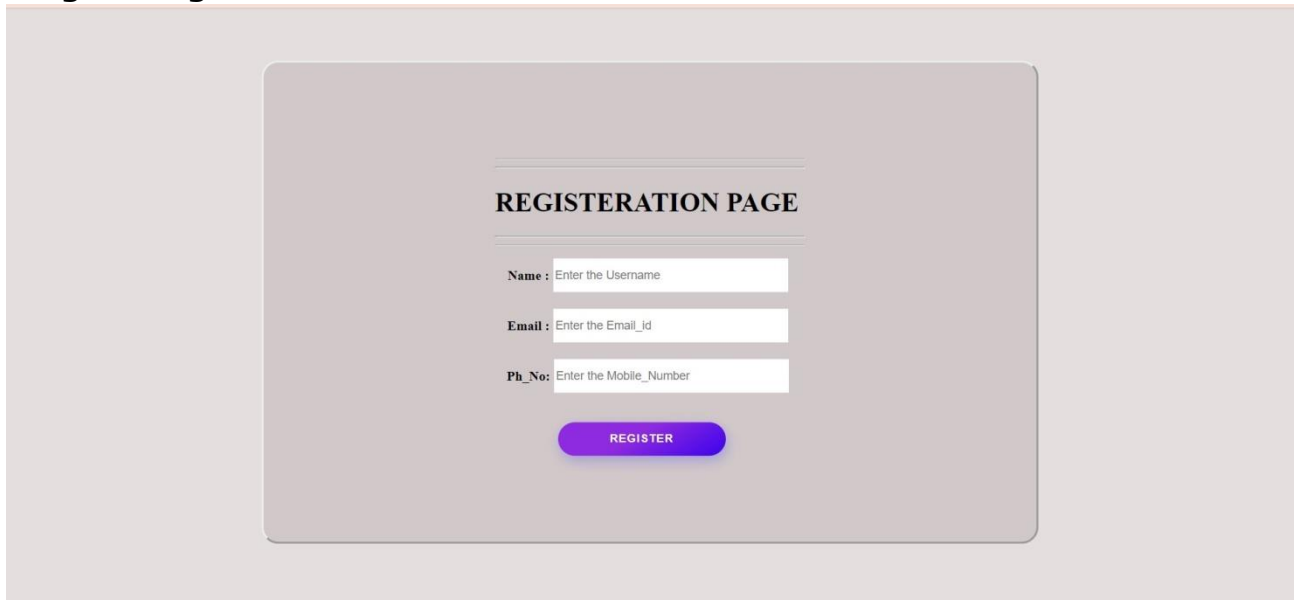
    return render_template('login.html', name='Home')

@app.route('/logout')
def logout():
    session.pop('email', None)
    return redirect(url_for('login'))
if __name__ == "__main__":
    app.run(debug=True)

```

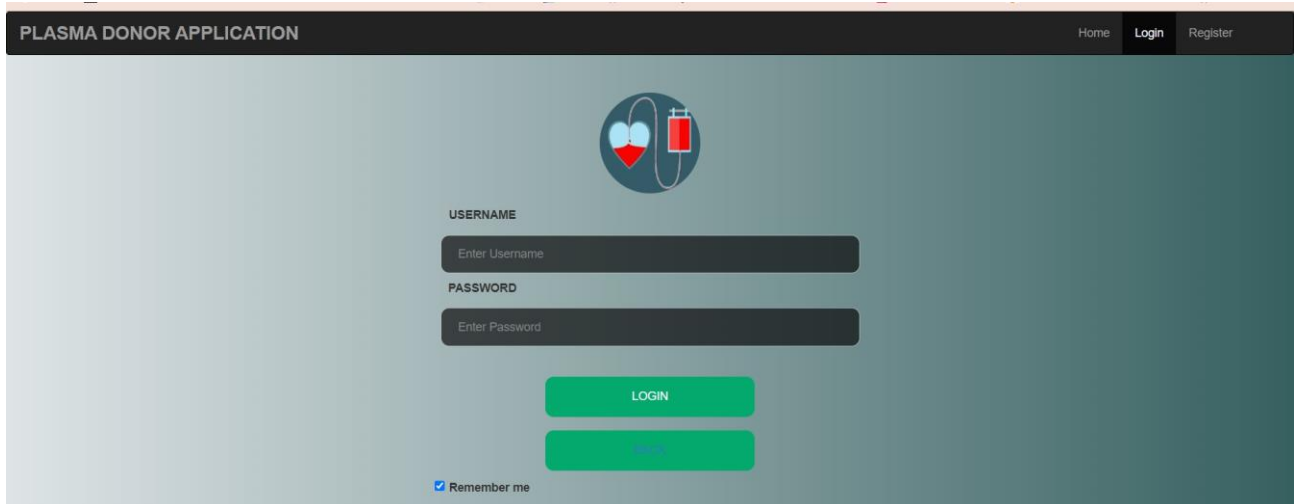
OUTPUT:

Registering:



A screenshot of a registration page. The page has a light gray background. In the center, there is a white rounded rectangle with a thin gray border. Inside this rectangle, the text "REGISTRATION PAGE" is centered in bold black font. Below the title, there are three input fields, each with a label and a placeholder text: "Name : Enter the Username", "Email : Enter the Email_id", and "Ph_No: Enter the Mobile_Number". Below these fields is a purple rounded button with the text "REGISTER" in white.

Logging in:



A screenshot of a login page for a "PLASMA DONOR APPLICATION". The page has a dark gray header with the title "PLASMA DONOR APPLICATION" on the left and navigation links "Home", "Login", and "Register" on the right. The main content area has a dark teal background. In the center, there is a circular logo featuring a heart with a red and white design. Below the logo, there are two input fields: "USERNAME" with the placeholder "Enter Username" and "PASSWORD" with the placeholder "Enter Password". Below these fields are two green rounded buttons: "LOGIN" and "Forgot". At the bottom left, there is a checkbox labeled "Remember me" which is checked.

Home page:



Success

You have registered successfully in Plasma donor Application
Thanks for visiting this website

Database:

KCV39089.USER_DETAILS

Back



Export to CSV

S_NO	USER_NAME	USER_EMAIL	ROLL_NO	PASSWORD
1	sasi kumar	sasi310@gmail.com	1	person@123
2	Tharini	tharinim25@gmail.com	2	person@123