

Assignment -2

DB and Flask

Student Name	R Keerthiya
Student Roll Number	621319104024
Maximum Marks	2 Marks

Question-1:

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

The screenshot shows the IBM Db2 on Cloud interface. On the left, the 'Data objects' pane lists a schema named 'STUDENT' containing a table 'USER'. The main editor shows a SQL script with the following statements:

```
1 create table users(  
2 SI_NO int generated by default as identity not null, Username varchar(200) not null, User_email varchar(200) not null, ro  
3 insert into users(Username, User_email, roll_no, password) values('anjana', 'anjana@gmail.com', 1, 'anj');  
4 insert into users(Username, User_email, roll_no, password) values('arthy', 'arthy@gmail.com', 2, 'art');  
5 insert into users(Username, User_email, roll_no, password) values('deepi', 'deepi@gmail.com', 3, 'deep');  
6 insert into users(Username, User_email, roll_no, password) values('keerthiya', 'keerthi@gmail.com', 4, 'keerthi');
```

The 'History' pane at the bottom displays the execution results of the insert statements:

Script	Date	Status	Runtime
insert into users(Username, User_email, roll_no, password) values('anjana', 'an...		✓	0.006 s
insert into users(Username, User_email, roll_no, password) values('arthy', 'art...		✓	0.009 s
insert into users(Username, User_email, roll_no, password) values('deepi', 'dee...		✓	0.005 s
insert into users(Username, User_email, roll_no, password) values('keerthiya', ...		✓	0.006 s

Insert Values:

The screenshot shows the IBM Db2 on Cloud interface. The SQL script in the editor includes an update statement:

```
7 update users set Username='puvi' where roll_no=34;  
8 desc users;
```

The 'History' pane shows the execution results, including the update statement which has a warning icon (yellow triangle) in the 'Status' column:

Script	Date	Status	Runtime
insert into users(Username, User_email, roll_no, password) values('anjana', 'an...		✓	0.006 s
insert into users(Username, User_email, roll_no, password) values('arthy', 'art...		✓	0.006 s
insert into users(Username, User_email, roll_no, password) values('deepi', 'dee...		✓	0.006 s
insert into users(Username, User_email, roll_no, password) values('keerthiya', ...		✓	0.006 s
update users set Username='puvi' where roll_no=34		⚠	0.005 s

User table:

The screenshot shows the IBM Db2 on Cloud interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The left sidebar has icons for 'Data objects', 'SQL', 'Tables', 'Views', 'MQTs', 'Aliases', and 'Nicknames'. The main area displays the table 'WDT07473.USERS'. A 'Back' button is in the top right. Below the table name is an 'Export to CSV' button. The table has five columns: SI_NO, USERNAME, USER_EMAIL, ROLL_NO, and PASSWORD. It contains six rows of data.

SI_NO	USERNAME	USER_EMAIL	ROLL_NO	PASSWORD
1	anjana	anjana@gmail.com	1	anj
2	anjana	anjana@gmail.com	1	anj
3	anjana	anjana@gmail.com	1	anj
4	arthy	arthy@gmail.com	2	art
5	deepi	deepi@gmail.com	3	deep
6	keerthiya	keerthi@gmail.com	4	keerthi

Question-2:

Perform update & delete queries with the table.

Update values:

The screenshot shows the IBM Db2 on Cloud interface with the SQL editor open. The editor contains the following SQL script:

```
1 create table users(  
2 SI_NO int generated by default as identity not null, Username varchar(200) not null, User_email varchar(200) not null, ro:  
3 insert into users(Username,User_email,roll_no,password) values('anjana','anjana@gmail.com',1,'anj');  
4 insert into users(Username,User_email,roll_no,password) values('arthy','arthy@gmail.com',2,'art');  
5 insert into users(Username,User_email,roll_no,password) values('deepi','deepi@gmail.com',3,'deep');  
6 insert into users(Username,User_email,roll_no,password) values('keerthiya','keerthi@gmail.com',4,'keerthi');  
7 update users set Username='puvi' where roll_no=34;  
8 desc users;
```

The 'History' tab is selected, showing a table with the following columns: Script, Date, Status, Runtime. The table contains five rows of execution history.

Script	Date	Status	Runtime
insert into users(Username,User_email,roll_no,password) values('anjana','an...		✓	0.006 s
insert into users(Username,User_email,roll_no,password) values('arthy','art...		✓	0.006 s
insert into users(Username,User_email,roll_no,password) values('deepi','dee...		✓	0.006 s
insert into users(Username,User_email,roll_no,password) values('keerthiya',...		✓	0.006 s
update users set Username='puvi' where roll_no=34		⚠	0.005 s

Delete values:

IBM Db2 on Cloud

Data objects

Saved objects

Filter objects

WDT07473

Tables

STUDENT

USER

Views

MQTs

Aliases

Nicknames

*Untitled - 1

SQL

1 create table users(
2 SI_NO int generated by default as identity not null, Username varchar(200) not null, User_email varchar(200) not null, ro:
3 insert into users(Username, User_email, roll_no, password) values('anjana', 'anjana@gmail.com', 1, 'anj');
4 insert into users(Username, User_email, roll_no, password) values('arthy', 'arthy@gmail.com', 2, 'art');
5 insert into users(Username, User_email, roll_no, password) values('deepi', 'deepi@gmail.com', 3, 'deep');
6 insert into users(Username, User_email, roll_no, password) values('keerthiya', 'keerthi@gmail.com', 4, 'keerthi');
7 update users set Username='puvi' where roll_no=34;
8 delete from users where Username='anjana';

History

Results

Find history

Script	Date	Status	Runtime
insert into users(Username, User_email, roll_no, password) values('anjana', 'an...		✓	0.003 s
insert into users(Username, User_email, roll_no, password) values('arthy', 'art...		✓	0.003 s
insert into users(Username, User_email, roll_no, password) values('deepi', 'dee...		✓	0.003 s
insert into users(Username, User_email, roll_no, password) values('keerthiya', ,...		✓	0.004 s
update users set Username='puvi' where roll_no=34		⚠	0.003 s
delete from users where Username='anjana'		✓	0.005 s

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

LHK92942.USER

Back

Export to CSV

SL_NO	USER_NAME	USER_EMAIL	ROLL_NO	PASSWORD
1	Sathiyananth	sakthiyananth@gmail.com	51	Ibm1234
2	Karthikeyan	karthi@gmail.com	22	karthi@22
3	Naveen	Naveen@gmail.com	37	naveen@37

Question-3:

Connect python code to db2.

```
import ibm_db
```

```
import bcrypt
```

```
conn =
```

```
ibm_db.connect("DATABASE=bludb;
```

```
HOSTNAME=9938aec0-8105-433e-
```

```
8bf9-
```

```
0fbb7e483086.c1ogj3sd0tgtu0lqde0
```

```
0.databases.appdomain.cloud;PORT
```

```
=32459;SECURITY=SSL;SSLServerCert
```

```
ificate=DigiCertGlobalRootCA.crt;PR
```

```
OTOCOL=TCPIP;UID=wdt07473;PWD
```

```
=khxUTQVy0OaDAOdc","")
```

Question-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:**app.py**

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

import ibm_db
import bcrypt

conn =

    ibm_db.connect("DATABASE=blu
    db;HOSTNAME=9938aec0-8105-
    433e-8bf9-
    0fbb7e483086.c1ogj3sd0tgtu0lq
    de00.databases.appdomain.clou
    d;PORT=32459;SECURITY=SSL;SS
    LServerCertificate=DigiCertGloba
    lRootCA.crt;PROTOCOL=TCPIP;UI
    D=wdt07473;PWD=khxUTQVy0O
    aDAOdc","")

# 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('index.html',na

me='Home')

@app.route("/register",methods=['

GET','POST'])

def register():

    if request.method == 'POST':

        email = request.form['email']

        name = request.form['name']

        rollNo = request.form['rollNo']

        password =

        request.form['password']

    if not email or not name or not

rollNo or not password:

        return

        render_template('register.html',

error='Please fill all fields')

```

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

```
query = "SELECT * FROM
user_details WHERE email=? OR
rollNo=?"
```

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

```
ibm_db.bind_param(stmt,1,emai
l)
```

```
ibm_db.bind_param(stmt,2,rollN
o)
```

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

```
isUser =
```

```
ibm_db.fetch_assoc(stmt)
```

```
if not isUser:
```

```
insert_sql = "INSERT INTO
user_details(EMAIL, NAME,
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, name)
```

```

        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',s
    uccess="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',err
            or='Please fill all fields')

            query = "SELECT * FROM

```



```
user_details WHERE email=?"
```

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

```
ibm_db.bind_param(stmt,1,emai  
l)
```

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

```
isUser =
```

```
ibm_db.fetch_assoc(stmt)  
print(isUser,password)
```

```
if not isUser:
```

```
    return
```

```
render_template('login.html',err  
or='Invalid Credentials')
```

```
isPasswordMatch =
```

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

```
if not isPasswordMatch:
```

```
    return
```

```
render_template('login.html',err  
or='Invalid Credentials')
```

```
session['email'] = isUser['EMAIL']
```

```
return redirect(url_for('home'))
```

```
return
```

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

Output:

Registration Page:

127.0.0.1:5000/register

Preplnsta : Prepare... GitHub - IBM-EPBL/... Ideation Phase - Go... Brainstorming- Idea... (1) New Messages! IBM-Project-2733-1... IBM-EPBL/IBM-Proj... IBM-Project-2717-1...

Register Form

E-mail:

Name:

Roll Number:

Password:

[Already have an account? Login](#)

Login Page:

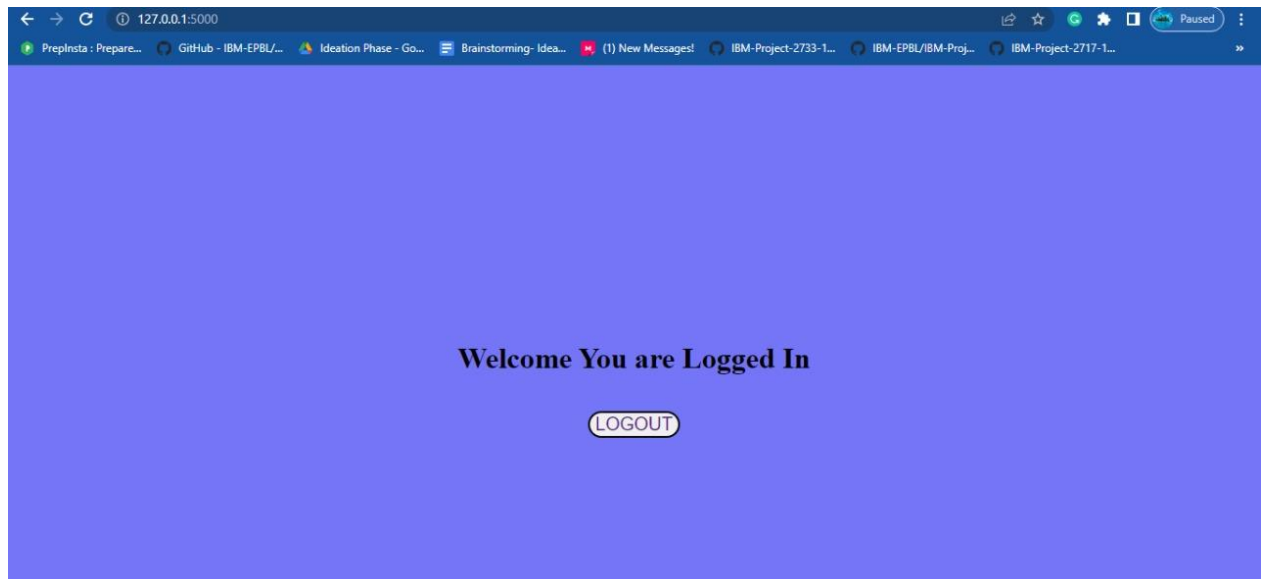
127.0.0.1:5000/login

Preplnsta : Prepare... GitHub - IBM-EPBL/... Ideation Phase - Go... Brainstorming- Idea... (1) New Messages! IBM-Project-2733-1... IBM-EPBL/IBM-Proj... IBM-Project-2717-1...

Login

[Don't have an account? Register](#)

Home Page:



Database:

