

ASSIGNMENT 2

Database connection

Date	12 October 2022
Student Name	Sivabalan T
Student Roll no	621319104053
Maximum Marks	2 Marks

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 shows a database named 'HVK46492'. The main editor area displays a SQL script to create a table named 'user_details'. The script is as follows:

```
1 CREATE TABLE user_details(  
2   sl_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 (sl_no)  
8 );
```

Below the script editor, the 'History' pane shows a table with columns: Script, Date, Status, and Runtime. It lists the execution of the 'Assignment_2' script on October 17, 2022, at 8:58:12 PM, with a status of 'Success' and a runtime of 0.180 s.

Inserting values :

The screenshot shows the IBM Db2 on Cloud interface with the SQL script editor displaying the following insert statements:

```
14  
15 insert into user_details(user_name,user_email,roll_no,password) values('Bala','bala2701@gmail.com',01,'stu@123');  
16 insert into user_details(user_name,user_email,roll_no,password) values('Ram','ram2701@gmail.com',02,'stu@123');  
17 insert into user_details(user_name,user_email,roll_no,password) values('Siva','siva2701@gmail.com',03,'stu@123');  
18
```

The 'History' pane shows the execution of these insert statements. It lists the execution of the 'Assignment_2' script on October 17, 2022, at 9:12:47 PM, with a status of 'Success' and a runtime of 0.060 s. It also shows the execution of the 'Assignment_2' script on October 17, 2022, at 9:11:53 PM, with a status of 'Success' and a runtime of 0.164 s.

User_details table:

The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected, and the table 'HVK46492.USER_DETAILS' is displayed. A 'Back' button is in the top right. Below the table name, there is an 'Export to CSV' button. The table itself has five columns: 'SL_NO', 'USER_NAME', 'USER_EMAIL', 'ROLL_NO', and 'PASSWORD'. It contains three rows of data.

SL_NO	USER_NAME	USER_EMAIL	ROLL_NO	PASSWORD
1	Bala	bala2701@gmail.com	1	stu@123
2	Ram	ram2701@gmail.com	2	stu@123
3	Siva	siva2701@gmail.com	3	stu@123

2. Perform update & delete queries with the table

The screenshot shows the IBM Db2 on Cloud web interface with the 'SQL' editor open. The 'Data objects' panel on the left shows the 'HVK46492' database. The 'SQL' editor contains the following query:

```
update user_details set user_name='Sivabalan' where roll_no=03;
```

The 'History' panel below the editor shows the execution history of the query:

Script	Date	Status	Runtime
Assignment_2	Oct 17, 2022 9:13:19 PM	✓ 1	0.020 s
update user_details set user_name='Sivabalan' where roll_no=03		✓	0.020 s

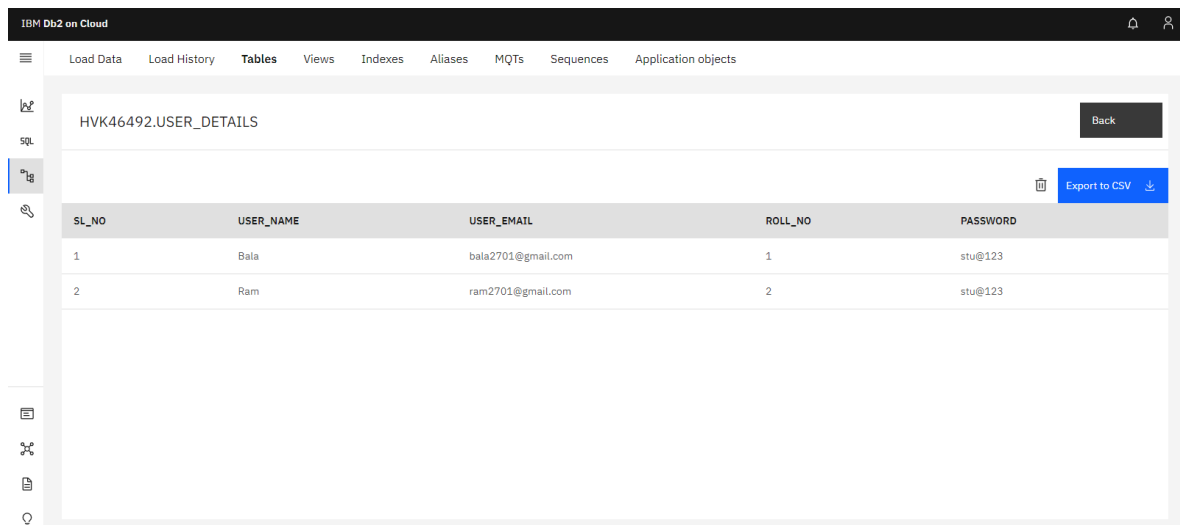
The screenshot shows the IBM Db2 on Cloud web interface with the 'SQL' editor open. The 'Data objects' panel on the left shows the 'HVK46492' database. The 'SQL' editor contains the following query:

```
delete from user_details where roll_no=03;
```

The 'History' panel below the editor shows the execution history of the query:

Script	Date	Status	Runtime
Assignment_2	Oct 17, 2022 9:15:13 PM	✓ 1	0.036 s
delete from user_details where roll_no=03		✓	0.036 s

Updated table:



The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected, and the table 'HVK46492.USER_DETAILS' is displayed. The table has five columns: 'SL_NO', 'USER_NAME', 'USER_EMAIL', 'ROLL_NO', and 'PASSWORD'. There are two rows of data. A 'Back' button is in the top right, and an 'Export to CSV' button is in the top right of the table area.

SL_NO	USER_NAME	USER_EMAIL	ROLL_NO	PASSWORD
1	Bala	bala2701@gmail.com	1	stu@123
2	Ram	ram2701@gmail.com	2	stu@123

3. Connect python code to db2

```
import ibm_db
import bcrypt

conn =ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;S
ECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UI
D=hvk46492;PWD=D3xaYotwDTyT0vm8","")
```

- 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 bcrypt

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-
4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;S
ECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UI
D=hvk46492;PWD=D3xaYotwDTyT0vm8","")
```

```

# 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:

Register

t.sivabalan2001@gmail.com

Siva

621319104053

Register

[Already have an account? Login](#)

Logging in:

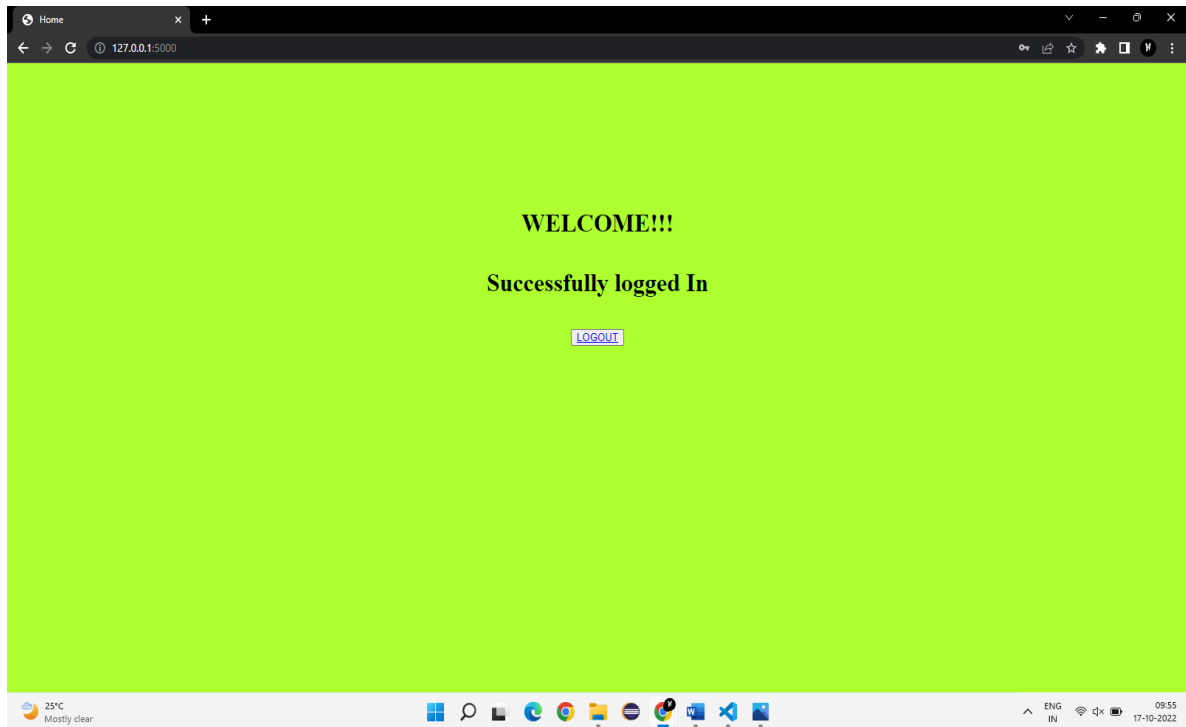
Login

t.sivabalan2001@gmail.com

Login

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

Home page:



Database:

