

## Assignment -2

Assignment Date	27 September 2022
Student Name	Mr. Jetson Cyrus J
Student Roll Number	9517201904060
Maximum Marks	2 Marks

### Question-1:

1. Create User table with user with email, username, roll number, password.
2. Perform UPDATE, DELETE Queries with user table
3. Connect python code to db2.
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:

#### Creating user table:

CREATE TABLE SAMPLE(name varchar(30), email varchar(30), password varchar(30))

The screenshot displays the IBM Db2 on Cloud web interface. At the top, a navigation bar includes the logo, a menu icon, and a notification bell. Below this, a red error banner states: "Error: Please check network connectivity then try again." with a "Show logs" link. The main workspace is divided into a left sidebar with icons for SQL, Tables, Views, and Functions, and a central area. The central area has a tab labeled "\*Untitled - 1" and a toolbar with icons for undo, redo, save, and other editing functions. A "Syntax assistant" toggle is visible. The SQL editor contains a script with 15 lines of code. The first 14 lines are:   
1 create table user(id int GENERATED BY DEFAULT AS IDENTITY NOT NULL, name varchar(30), email varchar(30), password varchar(30));  
2  
3  
4 alter table user add age int;  
5 select \* from hai;  
6 drop table user;  
7  
8 create table users(username varchar(30), email varchar(30), password varchar(30));  
9  
10 insert into users values('123','jetson@gmail.com', '12345');  
11 insert into users values('124','joshua@gmail.com', '12345');  
12 select \* from users;  
13  
14  
15 create table sample(name varchar(30), mail varchar(30), phn varchar(30));  
The 15th line is highlighted in blue. Below the editor is a "Run selected" button and a checkbox labeled "Remember my selection". On the right, a "Result" panel shows the execution of the 15th line: "create table sample(name varchar(30), mail v..." with a "Run time: 0.073 s" and "Status: Success | Affected rows: 0".

## Performing update, delete:

update sample set mail= 'jet@gmail.com' where name = 'jetson';

The screenshot shows the IBM Db2 on Cloud console interface. At the top, there's a navigation bar with tabs for 'You are signed in as 16194', 'IBM', 'Technical Training on Cloud App', 'Service Details - IBM Cloud', and 'IBM Db2 on Cloud'. Below this is a red error banner that says 'Error: Please check network connectivity then try again.' with a 'Show logs' link. The main area is divided into two panels. The left panel, titled '\*Untitled - 1', contains a SQL script with the following lines: 1 create table user(id int GENERATED BY DEFAULT AS IDENTITY NOT NULL, name varchar(30)); 2 alter table user add age int; 3 select \* from hai; 4 drop table user; 5 create table users(username varchar(30), email varchar(30), password varchar(30)); 6 insert into users values('123','jetson@gmail.com', '12345'); 7 insert into users values('124','joshua@gmail.com', '12345'); 8 select \* from users; 9 create table sample(name varchar(30), mail varchar(30), phn varchar(30)); 10 insert into sample values('jetson', 'jetson@gmail.com', '7951717434'); 11 select \* from sample; The right panel shows the execution results for the last statement, 'select \* from sample'. It indicates a successful execution with a run time of 0.535 s. Below this, there's a table with the following data: | NAME | MAIL | PHN | | jetson | jetson@gmail.com | 7951717434 |

The screenshot shows the IBM Db2 on Cloud console interface after executing an update statement. The top navigation bar is the same. The red error banner is still present. The left panel shows the same SQL script as before, but with an additional line at the bottom: 12 update sample set mail= 'jet@gmail.com' where name = 'jetson'; The right panel shows the execution results for the last statement, 'update sample set mail= 'jet@gmail.com' where name = 'jetson';'. It indicates a successful execution with a run time of 0.004 s. Below this, there's a table with the following data: | NAME | MAIL | PHN | | jetson | jet@gmail.com | 7951717434 |

delete from sample where name = 'jetson';

The screenshot displays the IBM Db2 on Cloud console interface. The top navigation bar shows the user is signed in as 16194. The main content area is divided into a script editor on the left and a results pane on the right.

**Script Editor:** The script is titled "Untitled - 1" and contains the following SQL statements:

```
1 create table user(id int GENERATED BY DEFAULT AS IDENTITY NOT NULL, name varchar(30));
2
3 alter table user add age int;
4 select * from hai;
5 drop table user;
6
7 create table users(username varchar(30), email varchar(30), password varchar(30));
8
9 insert into users values('123','jetson@gmail.com', '12345');
10 insert into users values('124','joshua@gmail.com', '12345');
11 select * from users;
12
13
14
15 create table sample(name varchar(30), mail varchar(30), phn varchar(30));
16 insert into sample values('jetson', 'jetson@gmail.com', '7951717434');
17 select * from sample;
18 update sample set mail= 'jet@gmail.com' where name = 'jetson';
19 delete from sample where name = 'jetson';
```

**Results Pane:** The results pane shows the execution of the SQL script. The first result set, "Result set 1", displays the output of the "select \* from sample" statement. The table has three columns: NAME, MAIL, and PHN. The data row shows "jetson", "jet@gmail.com", and "7951717434".

**Error Message:** Below the results pane, an error message is displayed: "Error: Please check network connectivity then try again." The error message is highlighted in red.

### Base.html:

```
<html>

<head>

  <link rel="stylesheet" href="static/css/main.css">

  {% block head %}

  {% endblock %}

</head>

<body>

  {% block body %}

  {% endblock %}

</body>

</html>
```

### Main.css:

```
.login {
  width: 60%;
  border-radius: 1rem;
  background: gray;
  padding: 10px;
  margin: auto;
  text-align: center;
  font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
}

.back{
  background-color: black;
}

.sub{
  background-color: black;
  color: white;
  border-radius: 1rem;
}

.box{
```

```
border-radius: 1rem;

background-color: black;

color: white;

}
```

#### Dashboard.html:

```
{% extends 'base.html' %}

{% block head %}

<title>Login</title>

{% endblock %}

{% block body %}

<br><br><br><br><br>  <br><br><br><br><br>

<div class="login">

  <h1>

    Welcome to dashboard.. {{ msg }}

  </h1>

  <hr>

</div>

{% endblock %}
```

#### Login.html:

```
{% extends 'base.html' %}

{% block head %}

<title>Login</title>

{% endblock %}

{% block body %}

<br><br><br><br><br><br><br>

<div class="login">

  <h1>

    Login

  </h1>

  <hr>

  <form action="/login" method="POST">
```

```

        <br>

        username: <input type="text" class="box" name="username" id="username"><br><br>

        Password: <input type="password" class="box" name="passwd" id="passwd"><br><br>

        <input type="submit" class="sub" value="Sign in">

    </form>

    Don't have account? <a href="/signup">Sign up</a>

</div>

{% endblock %}

```

### Signup.html:

```

{% extends 'base.html' %}

{% block head %}

<title>Login</title>

{% endblock %}

{% block body %}

<br><br><br><br><br> <br><br>

<div class="login">

    <h1>

        Sign Up

    </h1>

    <hr>

    <form action="/signup" method="POST">

        <br>

        username: <input type="text" class="box" name="username" id="username"><br><br>

        email: <input type="text" class="box" name="email" id="eamil"><br><br>

        Password: <input type="password" class="box" name="passwd" id="passwd"><br><br>

        <input type="submit" class="sub" value="Sign up"><br>

    </form>

    Already have an account? <a href="/login">Login</a>

</div>

{% endblock %}

```

### App.py:

```

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

import ibm_db

import re

import ibm_db_dbi

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;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=xyn12614;PWD=oXjOuam0AOYeLVKq
", "", "")

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

    global userid

    msg = ""

    if request.method == 'POST':

        username = request.form['username']

        password = request.form['passwd']

        sql = "select * from users 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'

        return render_template('dashboard.html', msg=msg)
    else:

        msg = 'Incorrect user credentials'

        return render_template('dashboard.html', msg=msg)
    else:

        return render_template('login.html')

@app.route('/signup', methods=["POST", "GET"])
def signup():
    if request.method == "POST":
        username = request.form['username']
        email = request.form['email']
        password = request.form['passwd']
        sql = "select * from users 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'^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z0-9]+', email):
            msg = 'invalid email'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain characters and numbers'
        else:
            insert_sql = "insert into users 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)

        print("successs")

        msg = "succesfully signed up"

        return render_template('dashboard.html', msg=msg)

    else:

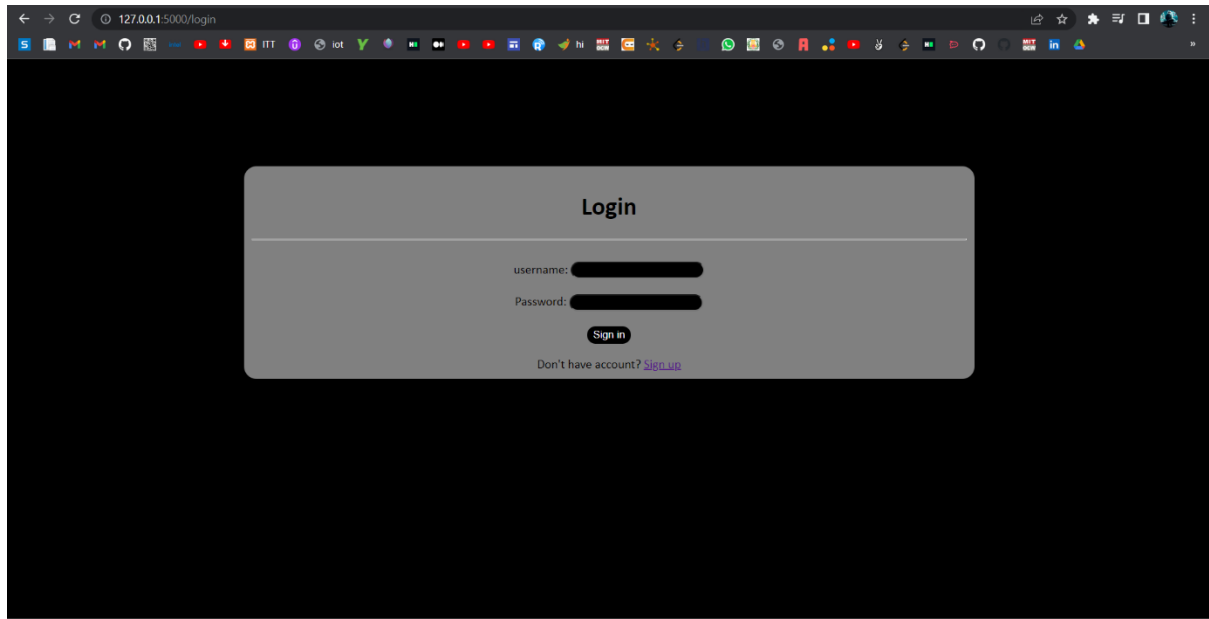
        return render_template('signup.html')

if __name__ == '__main__':

    app.run(debug=False)
```

Output:

Login:



127.0.0.1:5000/login

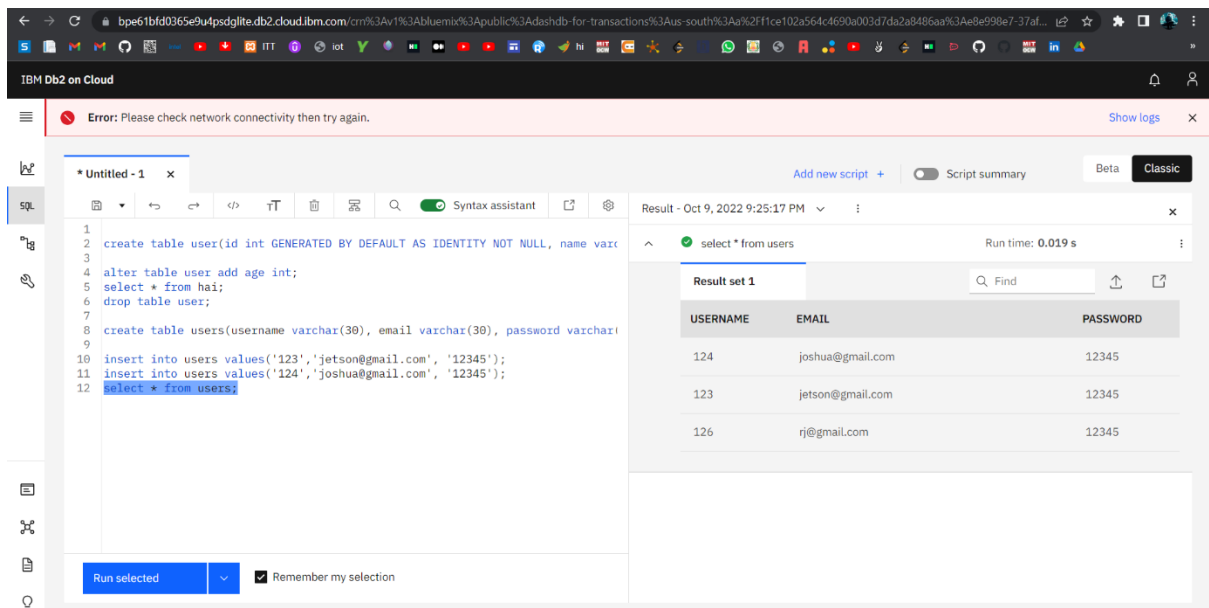
### Login

username:

Password:

[Sign in](#)

Don't have account? [Sign up](#)



IBM Db2 on Cloud

Error: Please check network connectivity then try again. [Show logs](#)

\*Untitled - 1

SQL

```
1 create table user(id int GENERATED BY DEFAULT AS IDENTITY NOT NULL, name varchar(30));
2
3
4 alter table user add age int;
5 select * from hai;
6 drop table user;
7
8 create table users(username varchar(30), email varchar(30), password varchar(30));
9
10 insert into users values('123','jetson@gmail.com', '12345');
11 insert into users values('124','joshua@gmail.com', '12345');
12 select * from users;
```

Result - Oct 9, 2022 9:25:17 PM

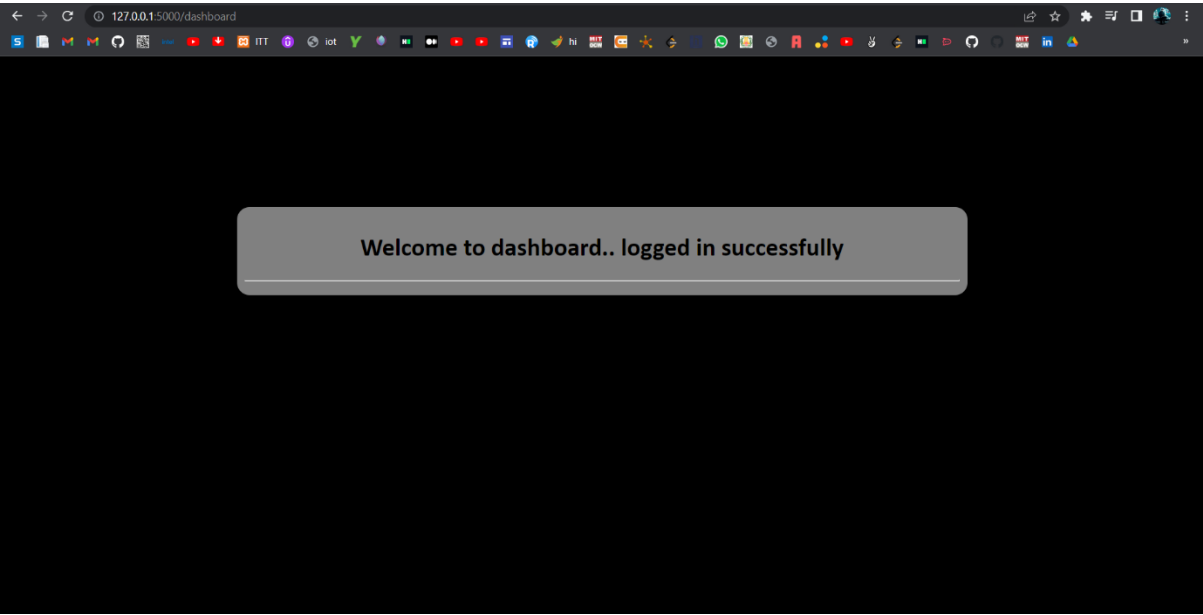
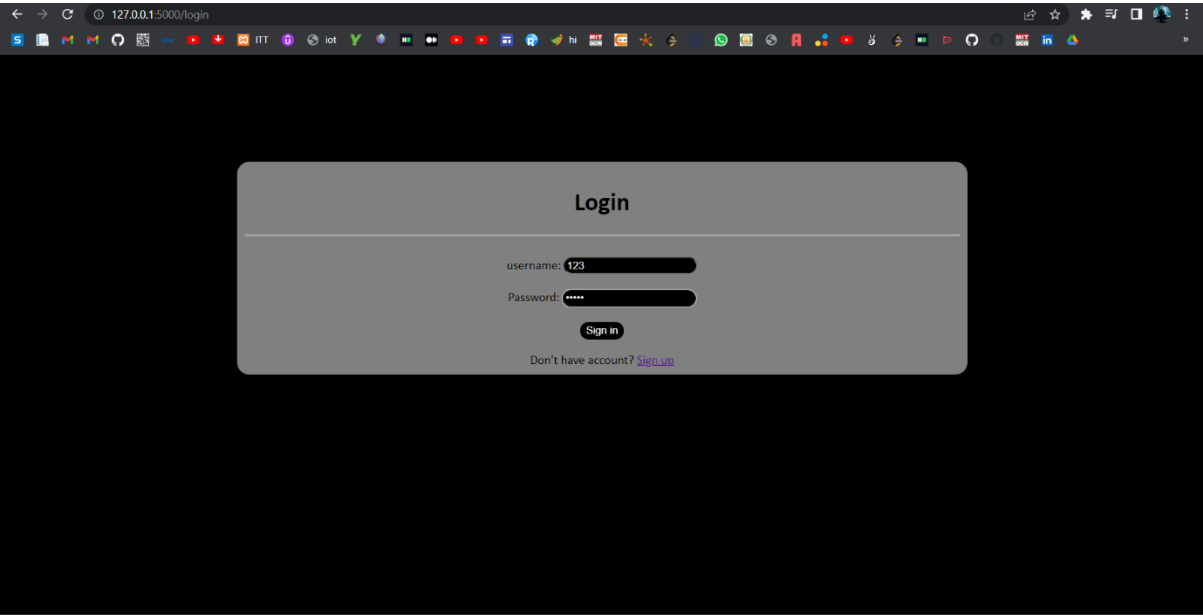
select \* from users Run time: 0.019 s

Result set 1

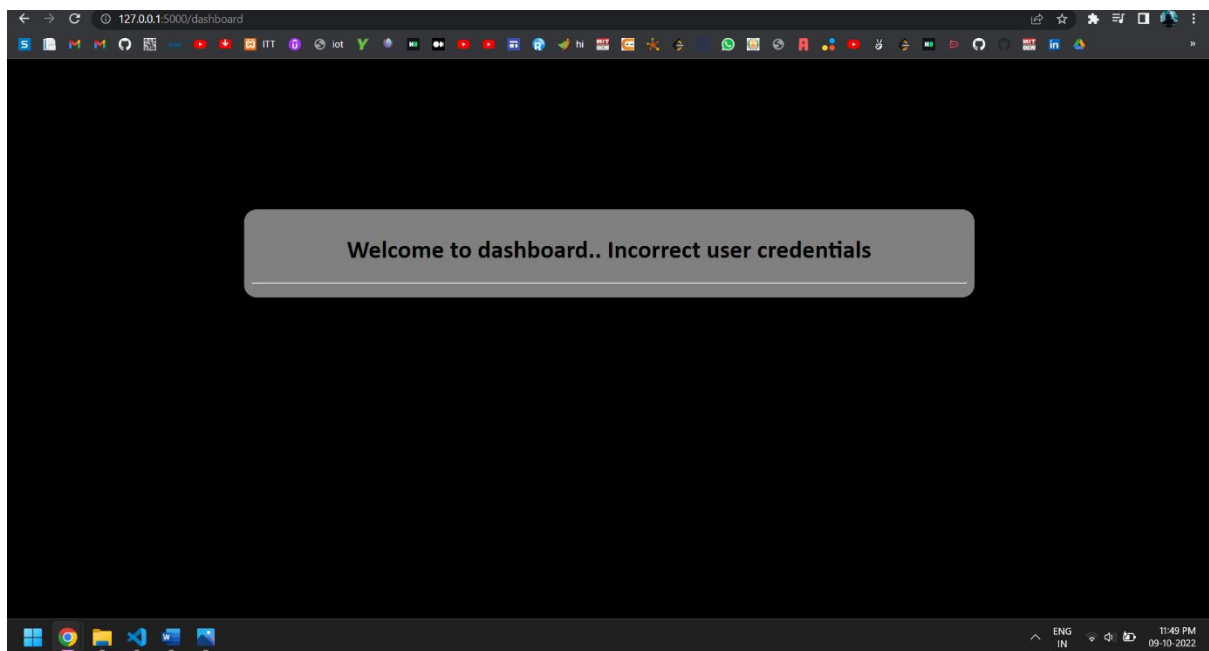
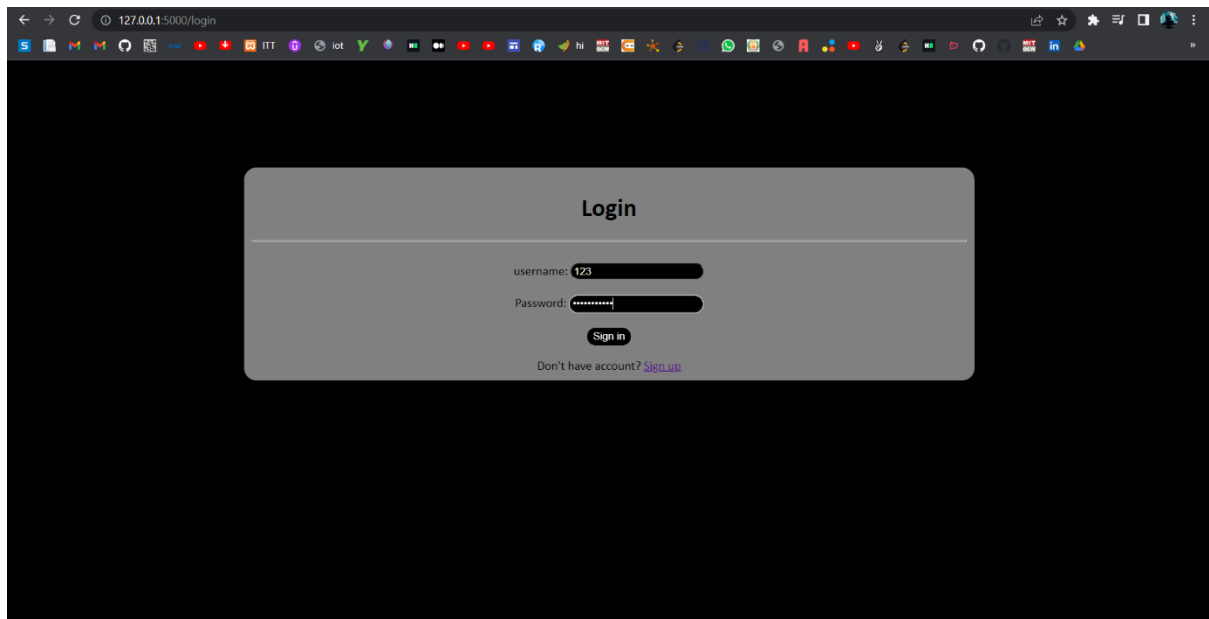
USERNAME	EMAIL	PASSWORD
124	joshua@gmail.com	12345
123	jetson@gmail.com	12345
126	rj@gmail.com	12345

Run selected ☐ Remember my selection

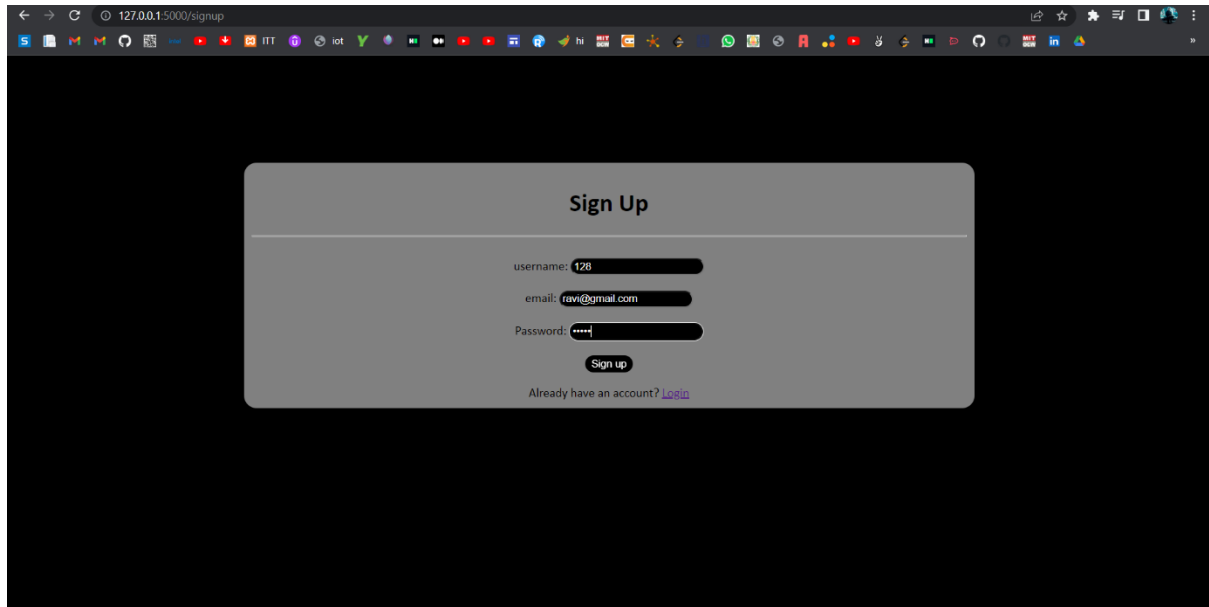
With correct credentials:



With wrong credentials:



Sign up:



Sign Up

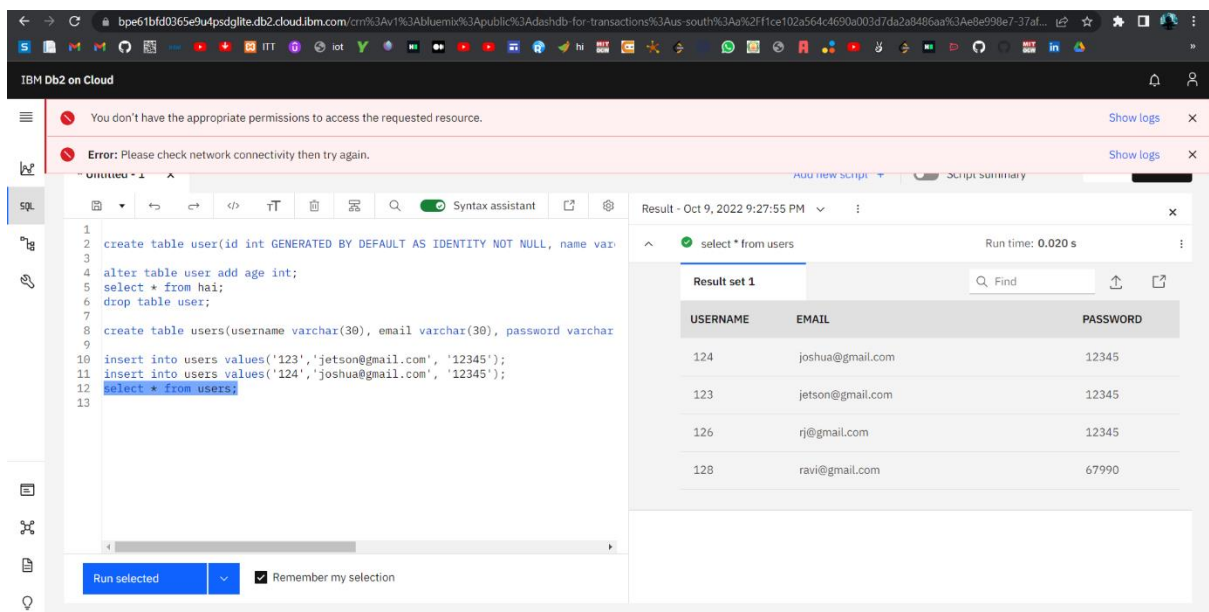
username: 128

email: ravi@gmail.com

Password: .....

Sign up

Already have an account? [Login](#)



IBM Db2 on Cloud

You don't have the appropriate permissions to access the requested resource. [Show logs](#)

Error: Please check network connectivity then try again. [Show logs](#)

SQL

```
1 create table user(id int GENERATED BY DEFAULT AS IDENTITY NOT NULL, name var
2
3
4 alter table user add age int;
5 select * from hai;
6 drop table user;
7
8 create table users(username varchar(30), email varchar(30), password varchar
9
10 insert into users values('123','jetson@gmail.com', '12345');
11 insert into users values('124','joshua@gmail.com', '12345');
12 select * from users;
13
```

Result - Oct 9, 2022 9:27:55 PM

select \* from users Run time: 0.020 s

Result set 1

USERNAME	EMAIL	PASSWORD
124	joshua@gmail.com	12345
123	jetson@gmail.com	12345
126	rj@gmail.com	12345
128	ravi@gmail.com	67990

Run selected Remember my selection