

Assignment -2

IBM CLOUD DB2

Assignment Date	29 September 2022
Student Name	Deepak H
Student Roll Number	311019205010
Maximum Marks	2 Marks

Questions:

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

IBM Db2 on Cloud

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

Find schemas or tables Refresh

Tables New table +

Name	Schema	Properties
USER	NNR87238	...

Total: 1, selected: 1

Table definition

USER No statistics available.

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	Y	32	0
EMAIL	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

View data

IBM Db2 on Cloud

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

NNR87238.USER Back

USERNAME	EMAIL	PASSWORD
----------	-------	----------

There is no data here yet

2. Perform UPDATE,DELETE Queries with user table.

UPDATE TABLE:-

The screenshot shows the IBM Db2 on Cloud SQL editor interface. On the left, the 'Data objects' pane shows a database named 'NNR87238'. The main editor area shows a script titled '*Untitled - 1' with the following SQL statement: `INSERT INTO USER VALUES ('ha','ha@gmail.com','1234')`. The 'Run all' button is visible. Below the editor, the 'History' tab shows the execution details of the script.

Script	Date	Status	Runtime
Untitled - 1	Oct 19, 2022 2:32:02 PM	✓ 1	0.007 s
INSERT INTO USER VALUES ('ha','ha@gmail.com','1234')		✓	0.007 s

The screenshot shows the IBM Db2 on Cloud SQL editor interface displaying the contents of the 'NNR87238.USER' table. The table has three columns: 'USERNAME', 'EMAIL', and 'PASSWORD'. The data row shows 'ha' as the username, 'ha@gmail.com' as the email, and '1234' as the password. The 'Export to CSV' button is visible.

USERNAME	EMAIL	PASSWORD
ha	ha@gmail.com	1234

DELETE TABLE:-

The screenshot shows the IBM Db2 on Cloud console interface. The top navigation bar includes tabs for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The 'Tables' tab is selected, displaying a table named 'NNR87238.USER'. The table has three columns: USERNAME, EMAIL, and PASSWORD. The data rows are as follows:

USERNAME	EMAIL	PASSWORD
de	de@gmail.com	123
ha	ha@gmail.com	1234

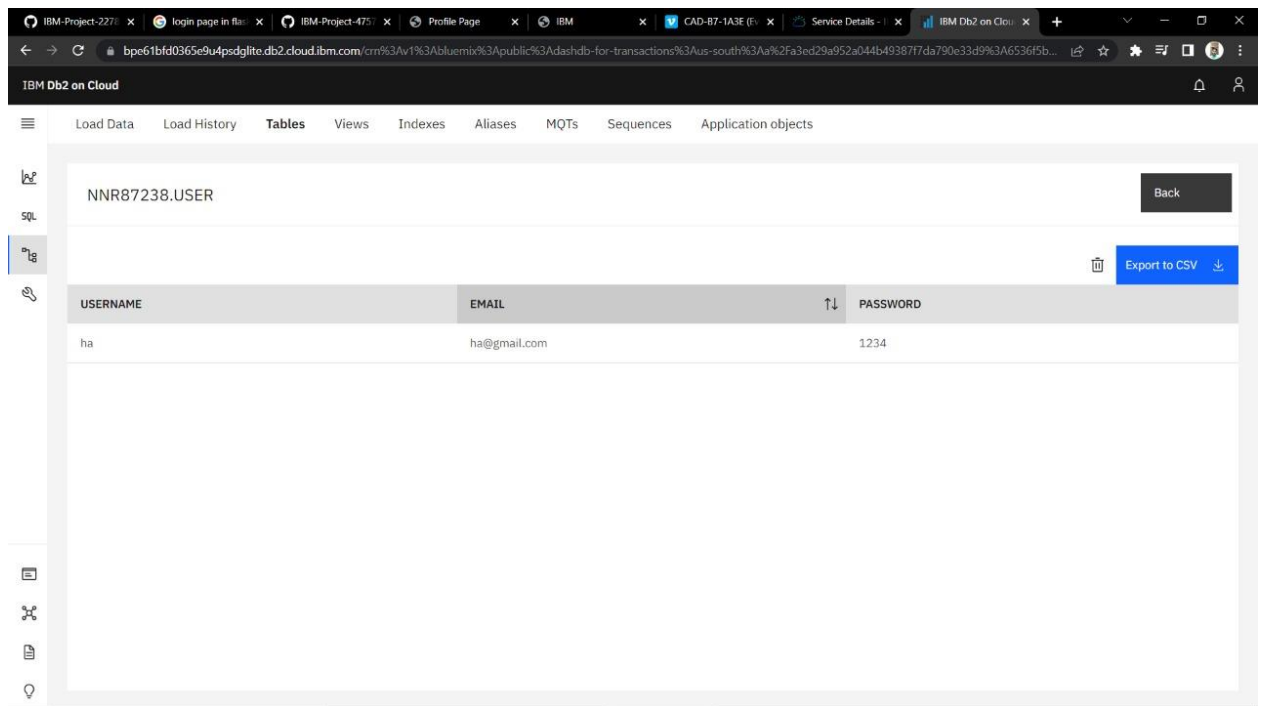
On the right side of the table, there is a 'Back' button and an 'Export to CSV' button.

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

```
1 DELETE FROM USER WHERE password = 123;
```

The 'History' tab is selected, showing a table with the following data:

Script	Date	Status	Runtime
Untitled - 1	Oct 19, 2022 2:54:24 PM	✓ 1	0.007 s
DELETE FROM USER WHERE password = 123		✓	0.007 s
Untitled - 1	Oct 19, 2022 2:53:30 PM	✗ 1	0.015 s
DELETE FROM USER WHERE username=de		✗	0.015 s
Untitled - 1	Oct 19, 2022 2:51:58 PM	✓ 1	0.007 s



3. Connect python code to db2.

NOTE:- Question 4 contains Question 3 answer

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

ANSWER:

app.js

```
const sign_in_btn = document.querySelector("#sign-in-btn");
const sign_up_btn = document.querySelector("#sign-up-btn");
const container = document.querySelector(".container");
```

```
sign_up_btn.addEventListener("click", () =>
{
  container.classList.add("sign-up-mode");
}
);
```

```
sign_in_btn.addEventListener("click", () =>
{
  container.classList.remove("sign-up-mode");
});
```

app.py

```
import ibm_db
import re
```

```
app = Flask(__name__)
app.secret_key = 'a'
```

```
conn = ibm_db.connect(
"DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-
d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;USERNAME=nnr87238;PASSWORD=PUMl
qXYukgiGmGq;SECURITY=SSL;SSLSERVERCERTIFICATE=DigiCertGlobalRootCA.crt;", "", "")
```

```

@app.route("/", methods=['GET', 'POST'])
def register():
    msg = ""
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        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-Z]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only 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)
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg=msg)

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

    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        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['USER_NAME']
            userid = account['USER_NAME']
            session['username'] = account['USER_NAME']

            msg = 'Logged in successfully !'
            return render_template('dashboard.html', msg=msg)
        else:
            msg = 'Incorrect username / password !'
            return render_template('login.html', msg=msg)

if __name__ == '__main__':

```

```
app.run(host='0.0.0.0')
```

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <script
      src="https://kit.fontawesome.com/64d58efce2.js"
      crossorigin="anonymous"
    ></script>
    <link rel="stylesheet" href="style.css" />
    <title>Sign in & Sign up Form</title>
  </head>
  <body>
    <div class="container">
      <div class="forms-container">
        <div class="signin-signup">
          <form action="#" class="sign-in-form">
            <h2 class="title">Sign in</h2>
            <div class="input-field">
              <i class="fas fa-user"></i>
              <input type="text" placeholder="Username" />
            </div>
            <div class="input-field">
              <i class="fas fa-lock"></i>
              <input type="password" placeholder="Password" />
            </div>
            <input type="submit" value="Login" class="btn solid" />
            <p class="social-text">Or Sign in with social platforms</p>
            <div class="social-media">
              <a href="#" class="social-icon">
                <i class="fab fa-facebook-f"></i>
              </a>
              <a href="#" class="social-icon">
                <i class="fab fa-twitter"></i>
              </a>
              <a href="#" class="social-icon">
                <i class="fab fa-google"></i>
              </a>
              <a href="#" class="social-icon">
                <i class="fab fa-linkedin-in"></i>
              </a>
            </div>
          </form>
          <form action="#" class="sign-up-form">
            <h2 class="title">Sign up</h2>
            <div class="input-field">
              <i class="fas fa-user"></i>
              <input type="text" placeholder="Username" />
            </div>
            <div class="input-field">
              <i class="fas fa-envelope"></i>
              <input type="email" placeholder="Email" />
            </div>
            <div class="input-field">
              <i class="fas fa-lock"></i>
              <input type="password" placeholder="Password" />
            </div>
            <input type="submit" class="btn" value="Sign up" />
          </form>
        </div>
      </div>
    </div>
  </body>
</html>
```

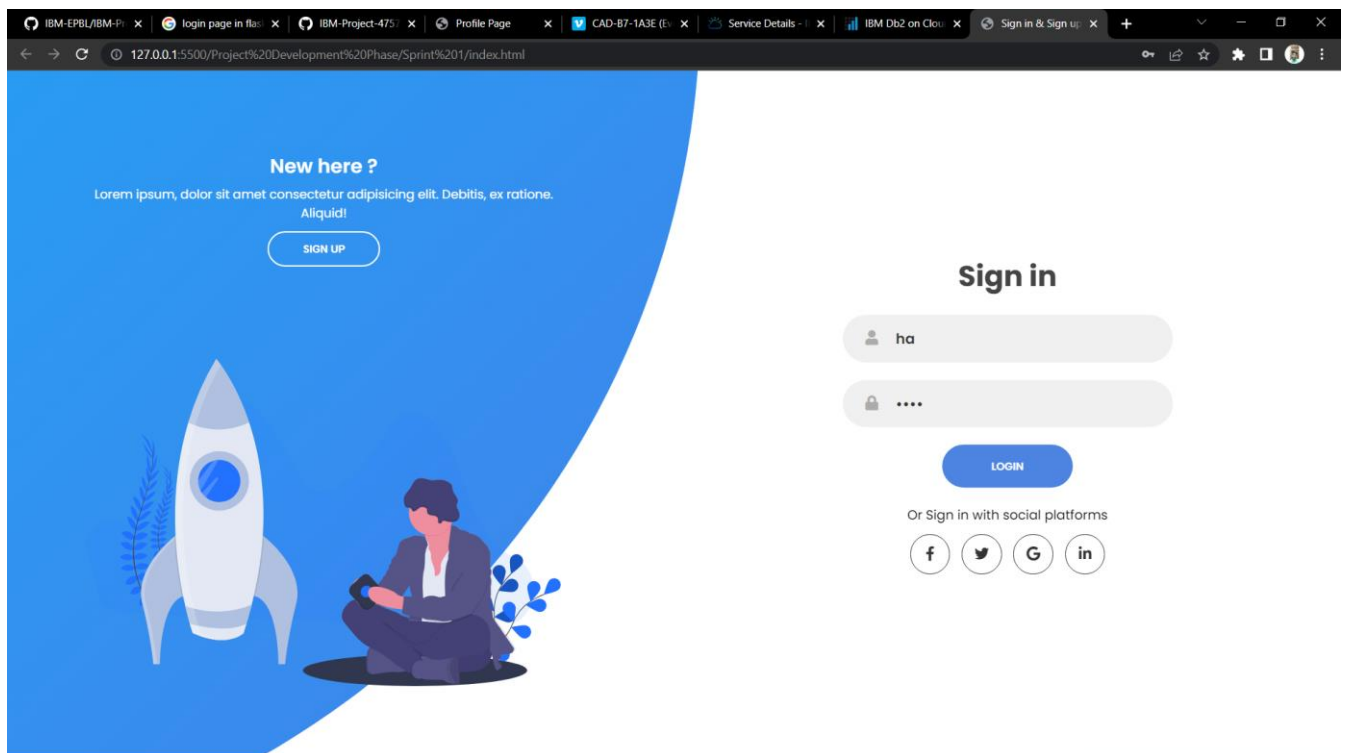
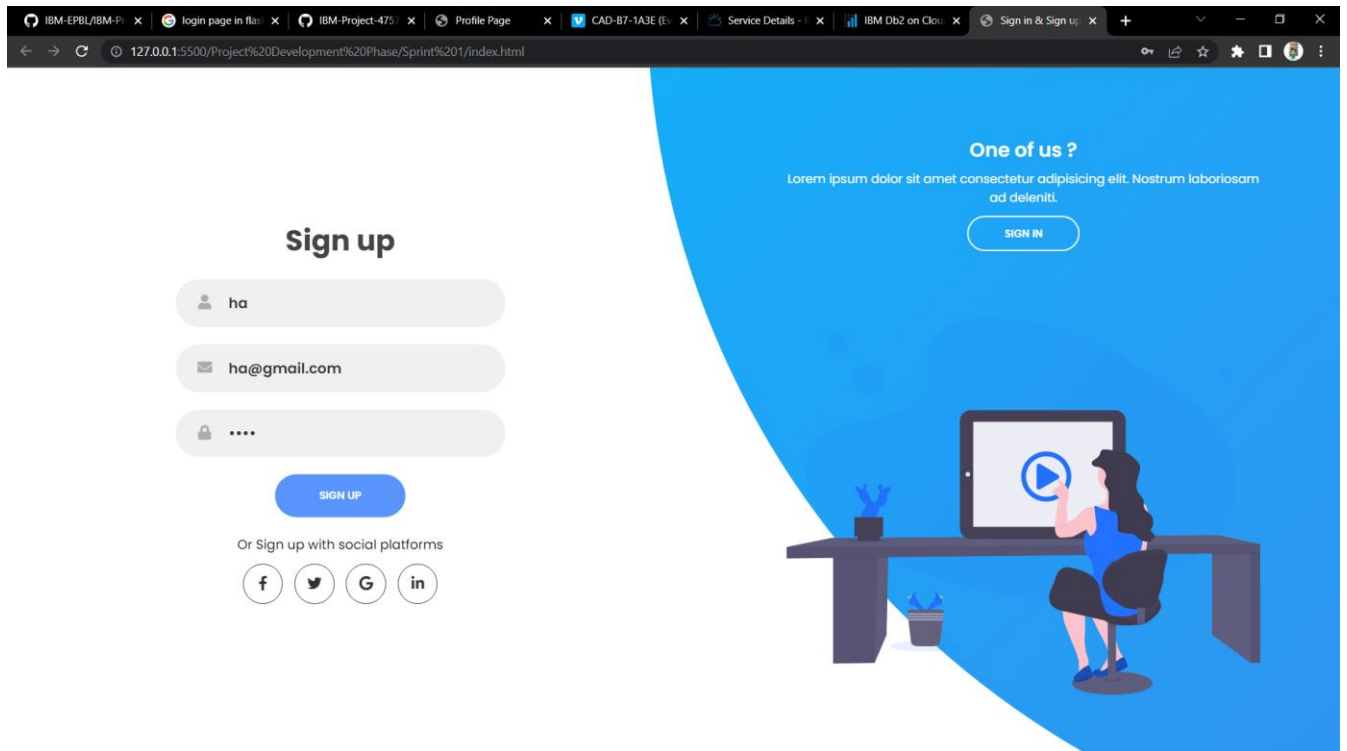
```

<p class="social-text">Or Sign up with social platforms</p>
<div class="social-media">
  <a href="#" class="social-icon">
    <i class="fab fa-facebook-f"></i>
  </a>
  <a href="#" class="social-icon">
    <i class="fab fa-twitter"></i>
  </a>
  <a href="#" class="social-icon">
    <i class="fab fa-google"></i>
  </a>
  <a href="#" class="social-icon">
    <i class="fab fa-linkedin-in"></i>
  </a>
</div>
</form>
</div>
</div>

<div class="panels-container">
  <div class="panel left-panel">
    <div class="content">
      <h3>New here ?</h3>
      <p>
        Lorem ipsum, dolor sit amet consectetur adipisicing elit. Debitis,
        ex ratione. Aliquid!
      </p>
      <button class="btn transparent" id="sign-up-btn">
        Sign up
      </button>
    </div>
    
  </div>
  <div class="panel right-panel">
    <div class="content">
      <h3>One of us ?</h3>
      <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Nostrum
        laboriosam ad deleniti.
      </p>
      <button class="btn transparent" id="sign-in-btn">
        Sign in
      </button>
    </div>
    
  </div>
</div>
</div>

<script src="app.js"></script>
</body>
</html>

```



IBM-EPBL/IBM-P...login page in fl...IBM-Project-475...Profile Page...CAD-B7-1A3E (E...Service Details -...IBM Db2 on Clou...Sign in & Sign up...

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fa3ed29a952a044b49387f7da790e33d9%3A6536f5b4-ead...

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

SQL

NNR87238.USER

Back

Export to CSV

USERNAME	EMAIL	PASSWORD
ha	ha@gmail.com	1234

NNR87238.USER

Back

Export to CSV

USERNAME	EMAIL	PASSWORD
ha	ha@gmail.com	1234