

ASSIGNMENT-2

IBM CLOUD DBS

Assignment Date	29 September 2022
Student Name	RANJITH KUMAR
Student Roll Number	951919CS082

Questions:

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

The screenshot displays the IBM Db2 on Cloud web 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 active, showing a search bar and a list of tables. A table named 'STUDENTS' is selected, and its definition is shown in the right-hand pane. The table definition pane lists the following columns:

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	Y	6	0
ROLLNO	INTEGER	Y		0
EMAIL ID	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

Below the table definition, there is a 'View data' button. The bottom status bar shows the system temperature as 23°C, the time as 22:50, and the date as 19-10-2022.

2. Perform UPDATE, DELETE Queries with user table.

INSERT TABLE:-

The screenshot shows the IBM Db2 on Cloud console interface. On the left, the 'Data objects' sidebar lists 'CKT48090'. The main editor area shows a SQL query: `insert into user values('ranjithkumar2002@gmail.com','RANJITH_KUMAR',1901081,123);`. Below the editor, the 'History' tab displays a table of executed queries.

Script	Date	Status	Runtime
Untitled - 1	Nov 9, 2022 11:15:52 AM	✓ 1	0.008 s
insert into user values('ranjithkumar2002@gmail.com','RANJITH_KUMAR',1901081,		✓	0.008 s

The screenshot shows the IBM Db2 on Cloud console with the 'Tables' tab selected. The table 'CKT48090.USER' is displayed. The table structure and data are as follows:

EMAIL	ROLL_NUMBER	USER_NAME	PASSWORD
ranjithkumar2002@gmail.com	RANJITH_KUMAR	1901081	123

UPDATE TABLE:-

The screenshot shows the IBM Db2 on Cloud SQL editor interface. The left sidebar displays the 'Data objects' tab with a search filter and a list of objects, including 'CKT48090'. The main editor area shows a script with two statements: an INSERT and an UPDATE. The UPDATE statement is highlighted, and the 'Run selected' button is visible. Below the script, the 'History' tab shows a table of executed scripts.

Script	Date	Status	Runtime
Untitled - 1	Nov 9, 2022 11:18:05 AM	✓ 1	0.008 s
UPDATE user SET password= 21052002 WHERE email ='ranjithkumar2002@gmail.com'		✓	0.008 s
Untitled - 1	Nov 9, 2022 11:15:52 AM	✓ 1	0.008 s
insert into user values('ranjithkumar2002@gmail.com','RANJITH_KUMAR',1901081,		✓	0.008 s

The screenshot shows the IBM Db2 on Cloud 'Tables' view. The table 'CKT48090.USER' is selected, and its data is displayed in a table format. The table has four columns: EMAIL, ROLL_NUMBER, USER_NAME, and PASSWORD. The data row shows the user 'ranjithkumar2002@gmail.com' with roll number 'RANJITH_KUMAR', user name '1901081', and password '123'. An 'Export to CSV' button is visible in the top right corner.

EMAIL	ROLL_NUMBER	USER_NAME	PASSWORD
ranjithkumar2002@gmail.com	RANJITH_KUMAR	1901081	123

DELETE TABLE:-

The screenshot shows the IBM Db2 on Cloud console interface. On the left, the 'Data objects' pane shows a table named 'CKT48090'. The main editor area displays a SQL script with three statements: an INSERT, an UPDATE, and a DELETE. The DELETE statement is highlighted: `DELETE FROM user WHERE email='ranjithkumar2002@gmail.com';`. Below the editor, the 'History' tab shows a table of executed scripts.

Script	Date	Status	Runtime
Untitled - 1	Nov 9, 2022 11:19:18 AM	✓ 1	0.008 s
DELETE FROM user WHERE email='ranjithkumar2002@gmail.com'		✓	0.008 s
Untitled - 1	Nov 9, 2022 11:18:05 AM	✓ 1	0.008 s
UPDATE user SET password= 21052002 WHERE email ='ranjithkumar2002@gmail.com'		✓	0.008 s
Untitled - 1	Nov 9, 2022 11:15:52 AM	✓ 1	0.008 s
insert into user values('ranjithkumar2002@gmail.com','RANJITH_KUMAR',1901081,		✓	0.008 s

The screenshot shows the IBM Db2 on Cloud console interface with the 'Tables' view selected. The table 'PTM66823.USER' is displayed. The table structure is shown with columns: EMAIL, USER NAME, ROLL NO, and PASSWORD. The table is currently empty, as indicated by the message 'There is no data here yet'.

EMAIL	USER NAME	ROLL NO	PASSWORD
-------	-----------	---------	----------

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

Answers App.py

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
app = Flask(__name__) app.secret_key

=

conn = ibm_db.connect(
    "DATABASE=Db2New;HOSTNAME=19af6446-6171-4641-8aba-
9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;USERNAM
E=kgb92696;PASSWORD=3IlWyGYU3bMIvwnX;SECURITY=SSL;SSLSERVERCERTIFICATE=DigiCer
tGlobalRootCA.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'^@+@[^@]+\.[^@]+', 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['USERNAME'] userid =
            account['USERNAME'] session['username'] =
            account['USERNAME'] msg = 'Logged in
            successfully !'

            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')
    # app.run(debug=True)

```

```

<html Lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Register Form</title>

  <!-- bootstrap css cdn -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.css">
    <!-- css stylesheet -->
    <link rel="stylesheet" href="Style.css">
    <!-- font styles cdn -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Alegreya&display=swap"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&display=swap"
" rel="stylesheet"> </head>

<body>
  <!-- bootstrap navbar -->
  <!-- navbar ends -->
  <!-- Login form -->
  <div class="login text-center mt-5">

    <form action="/" method="post">
      <h2 class="form-text-h2"> Register Form </h2>
      <div class="msg">{{ msg }}</div>
      <input type="text" class="form-control" name="username"
placeholder="Enter Your Username" id="username" required>
      <input type="email" class="form-control" name="email"
placeholder="Enter Your Email ID" id="email" required>
      <input type="password" class="form-control" name="password"
placeholder="Enter Your Password" id="password" required>
      <button type="submit" id="button" class="btn btn-primary"> Register
</button>

```

```

        <div class="note mt-3 text-center">
            <!--Register form -->
            <p> already have an account ? please login <a
href="/login">login! </a> </p>
        </div>
    </form>

</div>

</body>

</html>

```

Login.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>

    <link rel="icon" type="image/png" sizes="16x16" href="/assets/img/favicon-
32x32.png">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpsL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.css">

    <link rel="stylesheet" href="styleless.css">

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Alegreya&display=swap"
rel="stylesheet">

```



```

    <link
href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&display=swap"
rel="stylesheet">
</head>
<body>

    <div class="login text-center mt-5">

        <form action="/login" method="post">
            <h2 class="form-text-h2"> Login Form </h2>
            <div class="msg">{{ msg }}</div>
            <div class="form-group mb-3">
                <label class="label" for="username">Username</label>
                <input type="text" class="form-control" placeholder="Username"
name="username" required>
            </div>
            <div class="form-group mb-3">
                <label class="label" for="password">Password</label> <input
type="password" class="form-control"
placeholder="Password" name="password" required>
            </div>

            <button type="submit" id="button" class="btn btn-primary"> Login
</button>

            <div class="note mt-3 text-center">
                <!--Register form -->
                <p> Don't have an account yet? Click here to <a
href="/">register! </a> </p>
            </div>
        </form>
    </div>

</body>
</html>

```

Style.css

```

* { margin: 0;
padding: 0;
} body { color: #444444; background:linear-gradient(-45deg, #EE7752,
#E73C7E, #23A6D5, #23D5AB); background-size: 400% 400%; position: relative;
display: flex; flex-direction: column;

```

```

        align-items: center; justify-
content: center; animation: anime 10s
ease-in-out infinite; }
@keyframes anime{
    0%{ background-position: 0
        50%;
    }
    50%{ background-position:
        100% 50%;
    }
    100%{ background-position:
0 50%; } } form { display:
flex; flex-direction: column;
align-items: center; justify-
content: center; background-
color: white; width: 40%;
height: 450px; border-radius:
50px; margin: 100px; gap:
25px;
}

h2 { font-size:
    large; font-
weight:
    bolder;
}

input { width: 80%
    !important;
}

div { display: flex;
align-items: center;
justify-content:
center; flex-
direction: column;
width: 90%; }

.form-text-vk {
    font-size:
    small;

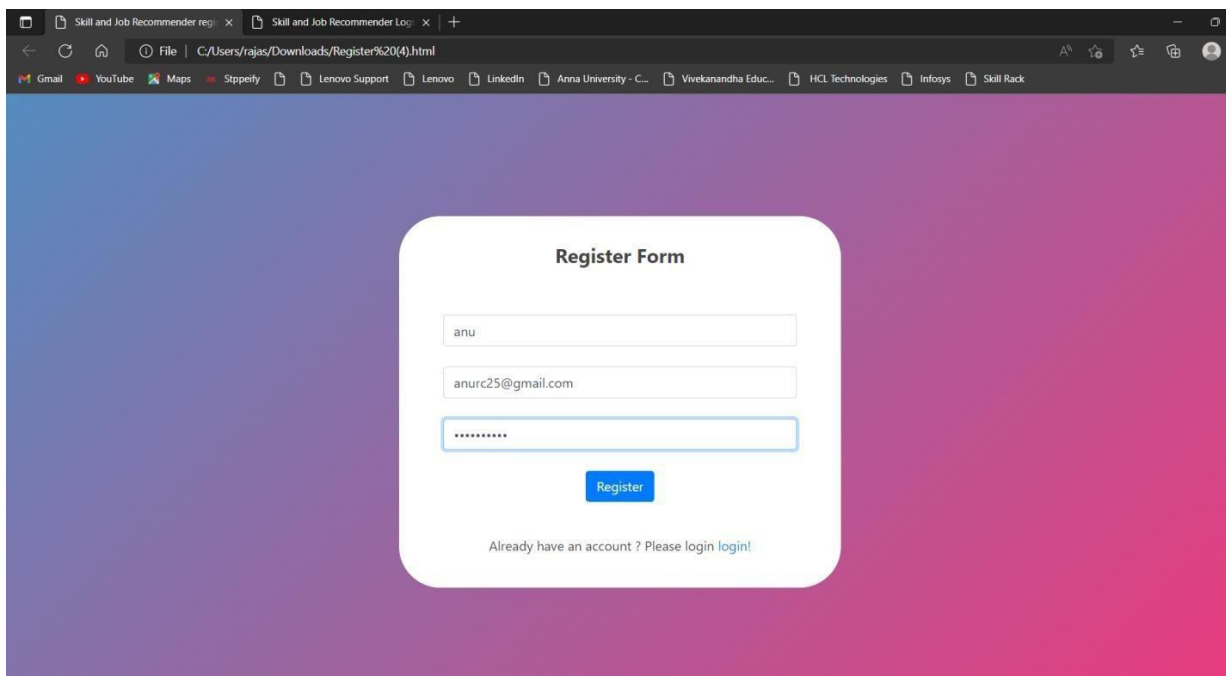
```

```
padding-bottom: 10px;
}

.form-text-h2 { font-size: x-large; padding-top: 10px;
}
```

OUTPUT:-

REGISTRATION PAGE:-



The screenshot shows a web browser window with two tabs. The active tab is titled 'Skill and Job Recommender Log...'. The address bar shows the file path 'C:/Users/rajas/Downloads/Register%20(4).html'. The browser's bookmark bar contains links to Gmail, YouTube, Maps, Stppety, Lenovo Support, Lenovo, LinkedIn, Anna University - C..., Vivekanandha Educ..., HCL Technologies, Infosys, and Skill Rack. The main content area features a registration form with a pink-to-purple gradient background. The form is titled 'Register Form' and contains three input fields: a username field with 'anu', an email field with 'anurc25@gmail.com', and a password field with masked characters. Below the fields is a blue 'Register' button. At the bottom of the form, there is a link: 'Already have an account ? Please login [login!](#)'.

LOGIN PAGE:-

Skill and Job Recommender reg: x Skill and Job Recommender Log: x +

File | C:/Users/rajas/Downloads/Login%20(2).html

Gmail YouTube Maps Stppeify Lenovo Support Lenovo LinkedIn Anna University - C... Vivekanandha Educ... HCL Technologies Infosys Skill Rack

Login Form

Username

anu

Password

Login

Don't have an account yet? Click here to [register!](#)

WELCOME PAGE:-

