

ASSIGNMENT-2

IBM CLOUD DBS

Assignment Date	29 September 2022
Student Name	Yoga Vignesh MK
Student Roll Number	951919CS118

Questions:

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

The screenshot shows the IBM Db2 on Cloud console interface. The 'Tables' tab is selected, and the 'STUDENTS' table is highlighted. The 'Table definition' panel on the right shows the following details:

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

The console also shows a search bar for schemas or tables, a 'New table' button, and a 'View data' button. The bottom status bar indicates the system is at 23°C and cloudy, with the date 19-10-2022 and time 22:50.

2. Perform UPDATE, DELETE Queries with user table.

INSERT TABLE:-

IBM Db2 on Cloud

Data objects

Saved objects

Filter objects

GBP44107

*Untitled - 1

1

2

insert into user values('yogavigneshmk@gmail.com','Yoga Vignesh MK',1901118,123);

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 9, 2022 10:07:51 AM	1	0.009 s
insert into user values('yogavigneshmk@gmail.com','Yoga Vignesh MK',1901118,-			0.009 s

28°C

AQI 66.29

ENG

IN

10:09 AM

09-11-2022

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

GBP44107.USER

Back

Export to CSV

EMAIL	USERNAME	ROLL_NO	PASSWORD
yogavigneshmk@gmail.com	Yoga Vignesh MK	1901118	123

28°C

Cloudy

ENG

IN

10:09 AM

09-11-2022

UPDATE TABLE:-

The screenshot shows the IBM Db2 on Cloud SQL editor interface. The left sidebar displays the 'Data objects' tree with a folder named 'GBP44107'. The main editor area shows a script titled '*Untitled - 1' with the following SQL statements:

```
1 insert into user values('yogavigneshmk@gmail.com','Yoga Vignesh MK',1901118,123);
2 UPDATE user SET password= 22102001 WHERE email ='yogavigneshmk@gmail.com';
```

Below the script, the 'History' tab shows a table of executed queries:

Script	Date	Status	Runtime
Untitled - 1	Nov 9, 2022 10:11:13 AM	✓ 1	0.006 s
UPDATE user SET password= 22102001 WHERE email ='yogavigneshmk@gmail.com'		✓	0.006 s
Untitled - 1	Nov 9, 2022 10:07:51 AM	✓ 1	0.009 s
insert into user values('yogavigneshmk@gmail.com','Yoga Vignesh MK',1901118,...		✓	0.009 s

The screenshot shows the IBM Db2 on Cloud SQL editor interface displaying the 'GBP44107.USER' table. The table structure is as follows:

EMAIL	USERNAME	ROLL_NO	PASSWORD
yogavigneshmk@gmail.com	Yoga Vignesh MK	1901118	22102001

The interface also includes a 'Back' button and an 'Export to CSV' button.

DELETE TABLE:-

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

SQL

GBP44107

```
1 insert into user values('yogavigneshmk@gmail.com','Yoga Vignesh MK',1901118,123);
2 UPDATE user SET password= 22102001 WHERE email ='yogavigneshmk@gmail.com';
3 DELETE FROM user WHERE email='yogavigneshmk@gmail.com';
```

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Nov 9, 2022 10:13:30 AM	✓ 1	0.006 s
DELETE FROM user WHERE email='yogavigneshmk@gmail.com'		✓	0.006 s
Untitled - 1	Nov 9, 2022 10:11:13 AM	✓ 1	0.006 s
UPDATE user SET password= 22102001 WHERE email ='yogavigneshmk@gmail.com'		✓	0.006 s
Untitled - 1	Nov 9, 2022 10:07:51 AM	✓ 1	0.009 s
insert into user values('yogavigneshmk@gmail.com','Yoga Vignesh MK',1901118,...		✓	0.009 s

28°C Cloudy

10:13 AM 09-11-2022

IBM Db2 on Cloud

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

GBP44107.USER

Back

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

There is no data here yet

28°C Cloudy

10:13 AM 09-11-2022

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/favicon32x32.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
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;
justifycontent: center;
backgroundcolor: white;
width: 40%; height: 450px;
border-radius:
50px; margin: 100px; gap:
25px;
}

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

input { width: 80%
    !important;
}

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

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

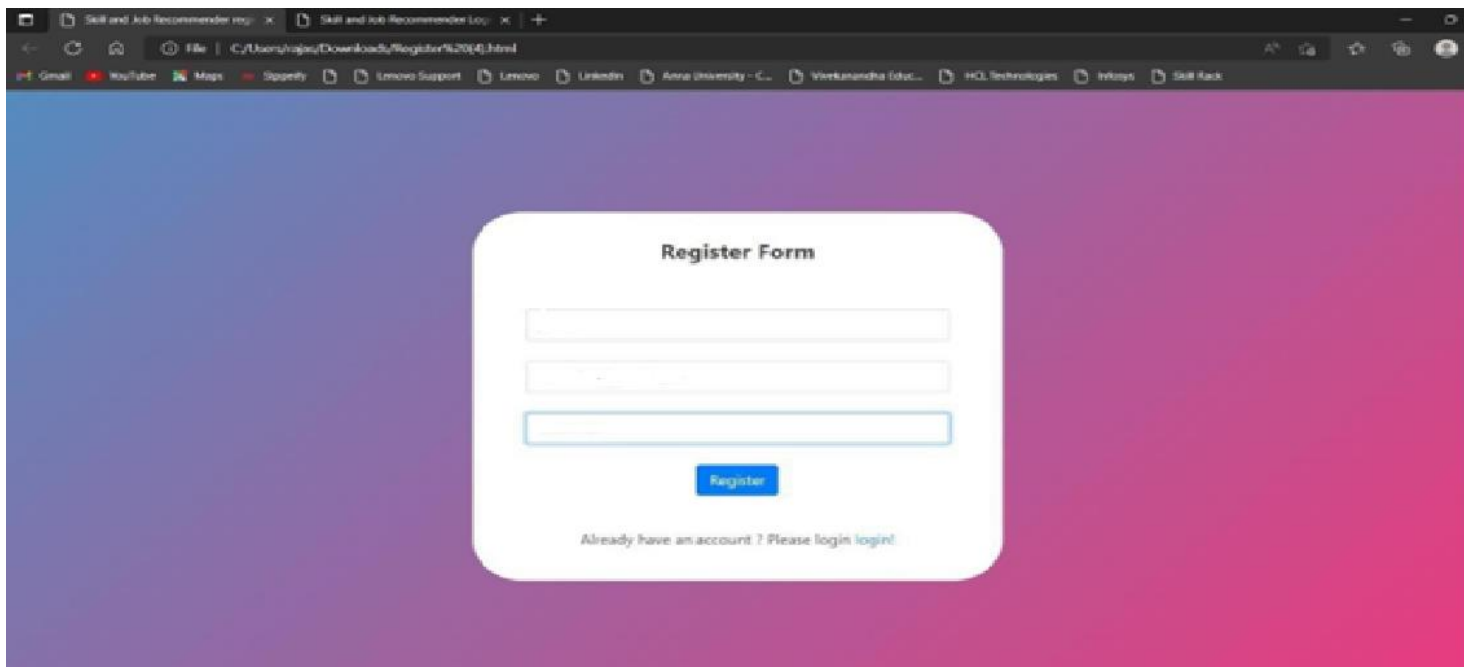
```

```
padding-bottom: 10px; }

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

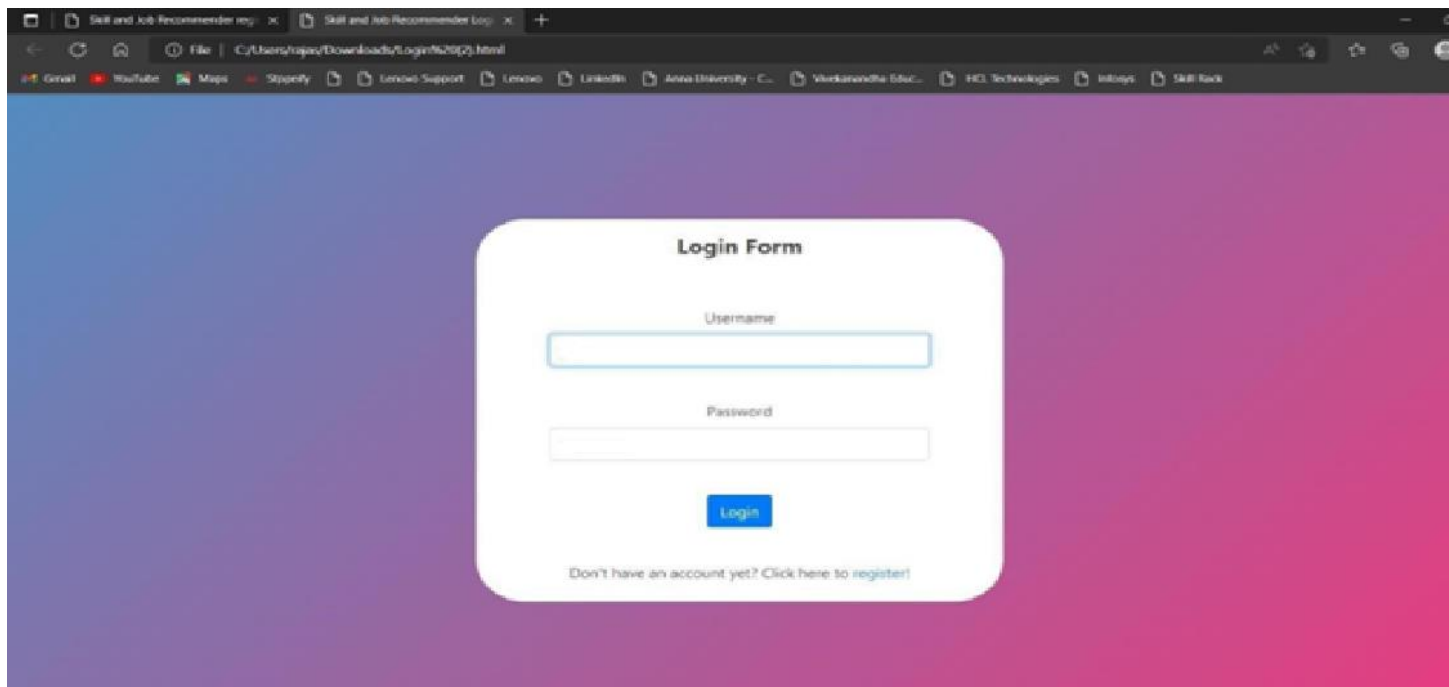
OUTPUT:-

REGISTRATION PAGE:-



The screenshot shows a web browser window with a registration form. The browser's address bar displays the file path: `C:/Users/rojan/Downloads/Register%20(4).html`. The form is centered on a purple-to-pink gradient background. It has a title "Register Form" and three input fields for "Name", "Email", and "Password". A blue "Register" button is positioned below the fields. At the bottom of the form, there is a link: "Already have an account ? Please login [login!](#)".

LOGIN PAGE:-



WELCOME PAGE:-

WELCOME TO DASHBOARD

WELCOME TO DASHBOARD