

**DHANALAKSHMI. S**

**311519104012**

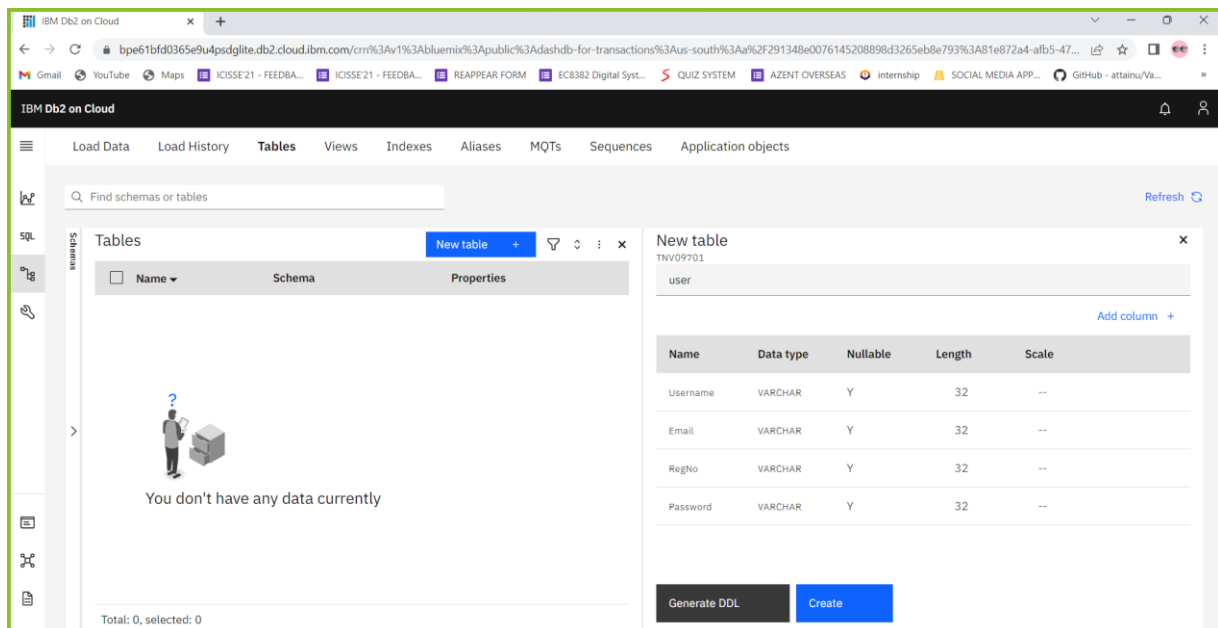
**[dhanalakshmi.sat@gmail.com](mailto:dhanalakshmi.sat@gmail.com)**

**PNT2022TMID27807**

## **ASSIGNMENT – II**

### **1. CREATE USER TABLE WITH USER WITH EMAIL , USERNAME , ROLL NUMBER , PASSWORD**

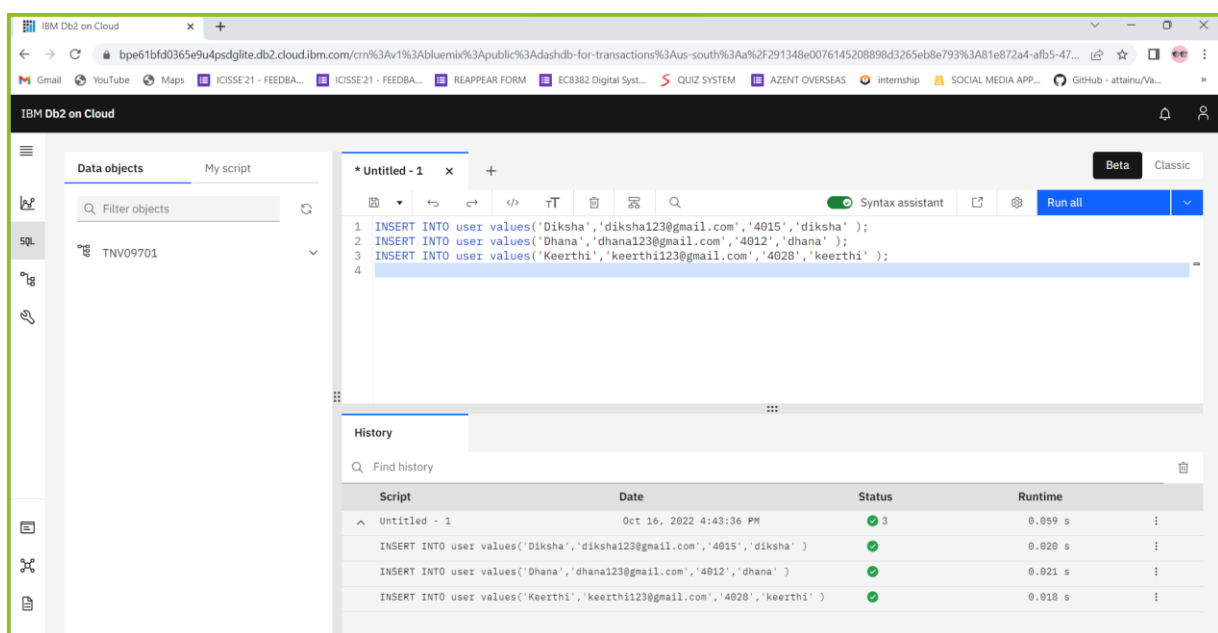
#### **OUTPUT**



The screenshot shows the IBM Db2 on Cloud console. The 'New table' dialog is open, showing the table name 'user' and the schema 'TNV09701'. The table structure is defined with the following columns:

Name	Data type	Nullable	Length	Scale
Username	VARCHAR	Y	32	--
Email	VARCHAR	Y	32	--
RegNo	VARCHAR	Y	32	--
Password	VARCHAR	Y	32	--

Buttons for 'Generate DDL' and 'Create' are visible at the bottom of the dialog.



The screenshot shows the IBM Db2 on Cloud console with the 'Data objects' tab selected. The 'My script' tab is active, displaying the following SQL script:

```
1 INSERT INTO user values('Diksha','diksha123@gmail.com','4015','diksha' );
2 INSERT INTO user values('Dhana','dhana123@gmail.com','4012','dhana' );
3 INSERT INTO user values('Keerthi','keerthi123@gmail.com','4028','keerthi' );
4
```

The 'Run all' button is visible. Below the script, the 'History' tab shows the execution results:

Script	Date	Status	Runtime
Untitled - 1	Oct 16, 2022 4:43:36 PM	3	0.059 s
INSERT INTO user values('Diksha','diksha123@gmail.com','4015','diksha' )		✓	0.020 s
INSERT INTO user values('Dhana','dhana123@gmail.com','4012','dhana' )		✓	0.021 s
INSERT INTO user values('Keerthi','keerthi123@gmail.com','4028','keerthi' )		✓	0.018 s

IBM Db2 on Cloud

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

TNV09701.USER Back

Export to CSV

USERNAME	EMAIL	REGNO	PASSWORD
Dhana	dhana123@gmail.com	4012	dhana
Diksha	diksha123@gmail.com	4015	diksha
Keerthi	keerthi123@gmail.com	4028	keerthi

## 2. PERFORM UPDATE, DELETE QUERIES WITH USER TABLE

### OUTPUT

IBM Db2 on Cloud

Data objects My script

Filter objects

TNV09701

\*Untitled - 1

**SQL Script:**

```

1 INSERT INTO user values('Diksha','diksha123@gmail.com','4015','diksha' );
2 INSERT INTO user values('Dhana','dhana123@gmail.com','4012','dhana' );
3 INSERT INTO user values('Keerthi','keerthi123@gmail.com','4028','keerthi' );
4 UPDATE user set password='anchana' WHERE username='Keerthi';

```

**History:**

Script	Date	Status	Runtime
Untitled - 1	Oct 16, 2022 4:47:12 PM	1	0.020 s
UPDATE user set password='anchana' WHERE username='Keerthi'			0.020 s

IBM Db2 on Cloud

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

TNV09701.USER

Back

Export to CSV

USERNAME	EMAIL	REGNO	PASSWORD
Dhana	dhana123@gmail.com	4012	dhana
Diksha	diksha123@gmail.com	4015	diksha
Keerthi	keerthi123@gmail.com	4028	anchana

IBM Db2 on Cloud

Data objects My script

Filter objects

TNV09701

Untitled - 1

```

1 INSERT INTO user values('Diksha','diksha123@gmail.com','4015','diksha' );
2 INSERT INTO user values('Dhana','dhana123@gmail.com','4012','dhana' );
3 INSERT INTO user values('Keerthi','keerthi123@gmail.com','4028','keerthi' );
4 UPDATE user set password='anchana' WHERE username='Keerthi';
5 DELETE FROM user WHERE username='Keerthi';

```

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 16, 2022 4:50:03 PM	1	0.019 s
DELETE FROM user WHERE username='Keerthi'			0.019 s

IBM Db2 on Cloud

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

TNV09701.USER

Back

Export to CSV

USERNAME	EMAIL	REGNO	PASSWORD
Dhana	dhana123@gmail.com	4012	dhana
Diksha	diksha123@gmail.com	4015	diksha

### 3. CONNECT PYTHON TO DB2

#### Register.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initialscale=1">
<style>
html {
    height: 100%;
    width: 100%;
}
body {
    background: url("https://images.unsplash.com/photo-1485802007047-
7774de7208e5?dpr=1&auto=compress,format&fit=crop&w=1800&h") no-repeat
center center fixed;
    background-size: cover;
    font-family: 'Droid Serif', serif;
}

#container {
    background: rgba(3, 3, 55, 0.5);
    width: 18.75rem;
    height: 25rem;
    margin: auto;
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
}
header {
    text-align: center;
    vertical-align: middle;
    line-height: 3rem;
    height: 3rem;
    background: rgba(3, 3, 55, 0.7);
    font-size: 1.4rem;
    color: #d3d3d3;
}
fieldset {
    border: 0;
    text-align: center;
}
input[type="submit"] {
```

```
background: #ff6600;
border: 0;
display: block;
width: 70%;
margin: 0 auto;
color: white;
padding: 0.7rem;
cursor: pointer;
}
input {
background: transparent;
border: 0;
border-left: 4px solid;
border-color: #ff6600;
padding: 10px;
color: white;
}

input[type="text"]:focus,
input[type="email"]:focus,
input[type="password"]:focus {
outline: 0;
background: #ff6600;
border-radius: 1.2rem;
border-color: transparent;
}

::placeholder {
color: #d3d3d3;
}

/*Media queries */

@media all and (min-width: 481px) and (max-width: 568px) {
#container {
margin-top: 10%;
margin-bottom: 10%;
}
}

@media all and (min-width: 569px) and (max-width: 768px) {
#container {
margin-top: 5%;
margin-bottom: 5%;
}
}
</style>
```

```

</head>
<body>
  <div id="container">
    <header>Registration Form</header>
    <form action="http://localhost:5000/register" method="POST">
      <fieldset>
        <br/>
        <input type="text" name="uname" id="uname" placeholder="name"
required autofocus>
        <br/><br/>
        <input type="email" name="email" id="email" placeholder="E-mail"
required>
        <br/><br/>
        <input type="text" name="regno" id="regno" placeholder="regno"
required>
        <br/><br/>
        <input type="password" name="pwd" id="pwd" placeholder="password"
required>
        <br/> <br/> <br/>
        <label for="submit"></label>
        <input type="submit" name="submit" id="submit" value="REGISTER">
      </fieldset>
    </form>
  </div>
</body>
</html>

```

### login.html

```

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initialscale=1">

<style>


html {

  height: 100%;

  width: 100%;

}

body {

```

```
background: url("https://images.unsplash.com/photo-1485802007047-7774de7208e5?dpr=1&auto=compress,format&fit=crop&w=1800&h") no-repeat center center fixed;
```

```
background-size: cover;
```

```
font-family: 'Droid Serif', serif;
```

```
}
```

```
#container {
```

```
background: rgba(3, 3, 55, 0.5);
```

```
width: 18.75rem;
```

```
height: 18rem;
```

```
margin: auto;
```

```
position: absolute;
```

```
top: 0;
```

```
left: 0;
```

```
right: 0;
```

```
bottom: 0;
```

```
}
```

```
header {
```

```
text-align: center;
```

```
vertical-align: middle;
```

```
line-height: 3rem;
```

```
height: 3rem;
```

```
background: rgba(3, 3, 55, 0.7);
```

```
font-size: 1.4rem;
```

```
color: #d3d3d3;
```

```
}
```

```
fieldset {
```

```
border: 0;
```

```
text-align: center;
```

```
}
```

```
input[type="submit"] {
```

```
background: #ff6600;
border: 0;
display: block;
width: 70%;
margin: 0 auto;
color: white;
padding: 0.7rem;
cursor: pointer;
}
input {
background: transparent;
border: 0;
border-left: 4px solid;
border-color: #ff6600;
padding: 10px;
color: white;
}
input[type="text"]:focus,
input[type="email"]:focus,
input[type="password"]:focus {
outline: 0;
background: #ff6600;
border-radius: 1.2rem;
border-color: transparent;
}
::placeholder {
color: #d3d3d3;
}
/*Media queries */
@media all and (min-width: 481px) and (max-width: 568px) {
```



```
#container {
    margin-top: 10%;
    margin-bottom: 10%;
}
}

@media all and (min-width: 569px) and (max-width: 768px) {
    #container {
        margin-top: 5%;
        margin-bottom: 5%;
    }
}

</style>
</head>
<body>
    <div id="container">
        <header>Login Form</header>
        <form action="http://localhost:5000/login" method="POST">
            <fieldset>
                <br/>
                <input type="text" name="uname" id="uname" placeholder="name" required
autofocus>
                <br/><br/>
                <input type="password" name="pwd" id="pwd" placeholder="password" required>
                <br/> <br/> <br/>
                <label for="submit"></label>
                <input type="submit" name="submit" id="submit" value="REGISTER">
            </fieldset>
        </form>
    </div>
</body>
</html>
```

## **app.py**

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

```
import ibm_db
```

```
import re
```

```
app=Flask(__name__)
```

```
app.secret_key='a'
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=tnv09701;PWD=PYHoZDtrj0Z92eb7";",",")
```

```
@app.route('/')
```

```
def home():
```

```
    return render_template('register.html')
```

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

```
def login():
```

```
    global userid
```

```
    msg=" "
```

```
    if request.method=="POST":
```

```
        username = request.form['uname']
```

```
        password = request.form['pwd']
```

```
        sql = "SELECT * FROM User 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 redirect(url_for('welcome', username=username))

else: msg = "Incorrect Username/Password"

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

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

def registration():

    msg = " "

    if request.method=="POST":

        username = request.form['uname']

        email = request.form['email']

        password = request.form['pwd']

        regno = request.form['regno']

        sql="SELECT * FROM user 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 = "Username must contain only alphabets and numbers."

        else:

```

```

insert_sql = "INSERT INTO user 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,regno)

ibm_db.bind_param(prepare_stmt,4,password)

ibm_db.execute(prepare_stmt)

msg = "You have successfully registered."

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

elif request.method == 'POST': msg="Please fill out the form."

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

@app.route('/welcome/<username>')

def welcome(username):

    return "Welcome %s!" %username

if __name__=="__main__":

    app.run(host='0.0.0.0')

```

## OUTPUT

```

C:\Users\DHANALAKSHMI\OneDrive\Documents\FLASK>app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.152:5000
Press CTRL+C to quit
127.0.0.1 - - [16/Oct/2022 17:58:09] "GET / HTTP/1.1" 200 -
False
127.0.0.1 - - [16/Oct/2022 17:58:28] "POST /register HTTP/1.1" 200 -
{'USERNAME': 'abc123', 'EMAIL': 'abc123@gmail.com', 'REGNO': '1234', 'PASSWORD': '12345'}
127.0.0.1 - - [16/Oct/2022 17:58:40] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [16/Oct/2022 17:58:40] "GET /welcome/abc123 HTTP/1.1" 200 -
127.0.0.1 - - [16/Oct/2022 18:01:24] "GET / HTTP/1.1" 200 -
False
127.0.0.1 - - [16/Oct/2022 18:03:25] "POST /register HTTP/1.1" 200 -
{'USERNAME': 'anchana', 'EMAIL': 'anchu123@gmail.com', 'REGNO': '1456', 'PASSWORD': '12345'}
127.0.0.1 - - [16/Oct/2022 18:04:04] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [16/Oct/2022 18:04:04] "GET /welcome/anchana HTTP/1.1" 200 -

```

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**

## OUTPUT

