

Assignment -2

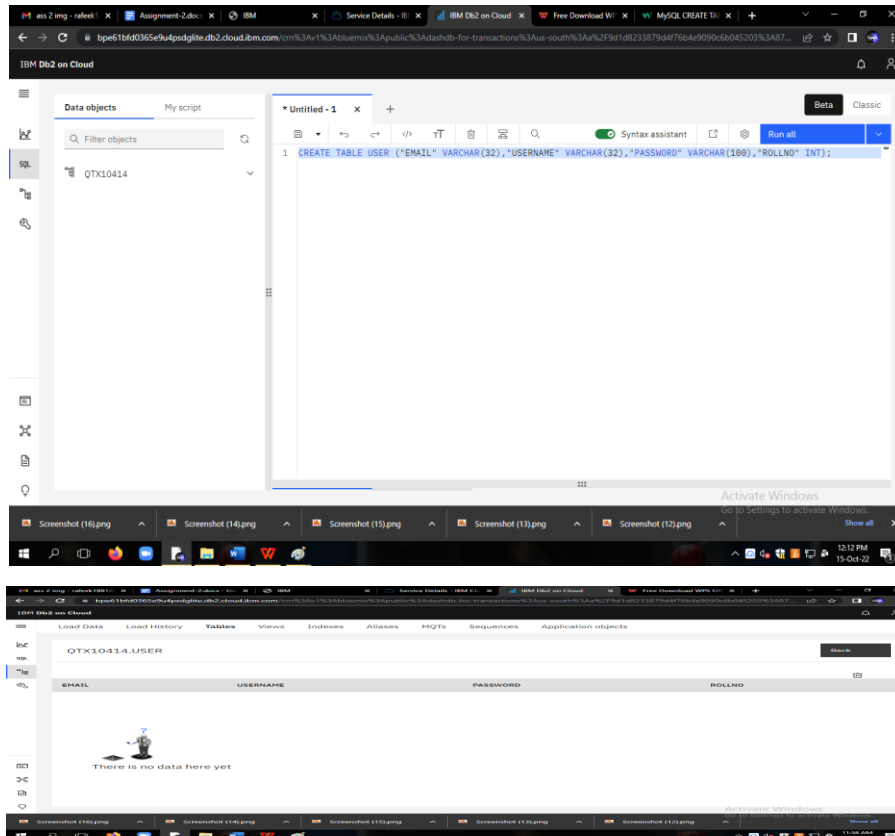
Connect with Database Assignment

Assignment Date	09 October 2022
Student Name	S.Mohammed Rafeek
Student Roll Number	621319104030
Maximum Marks	2 Marks

Question-1:

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

z



2. Perform UPDATE, DELETE Queries with User table

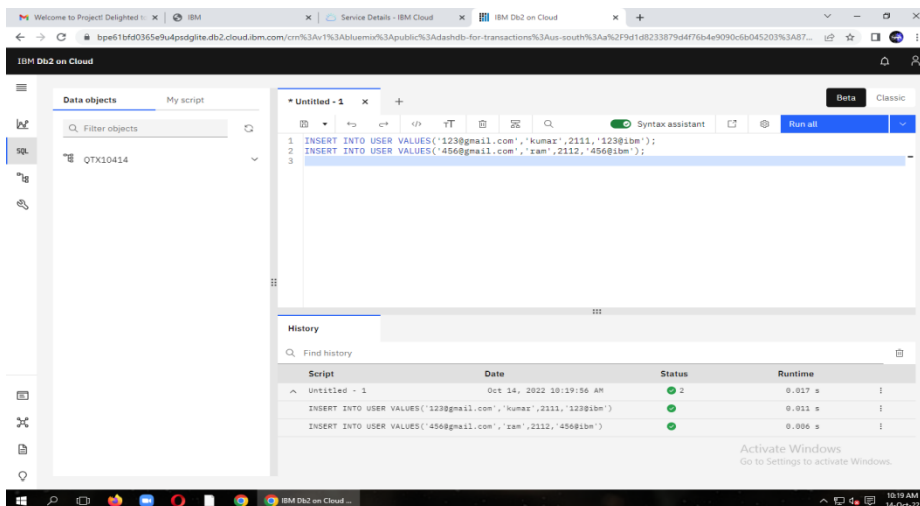
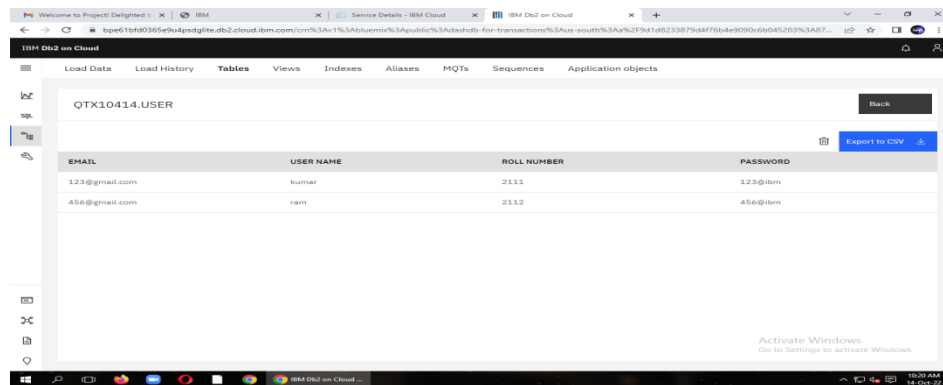


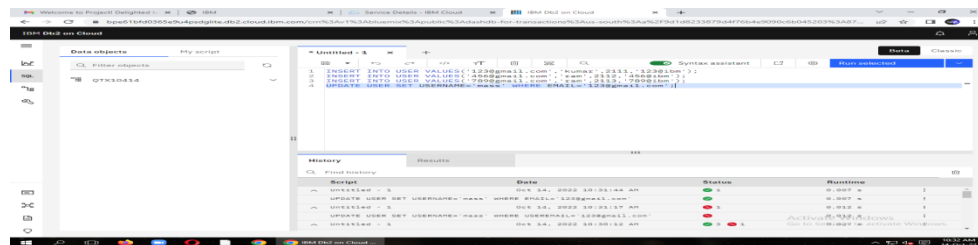
Table View:



The screenshot shows the IBM Db2 on Cloud interface. The 'Tables' tab is selected, displaying a table named 'QTX10414.USER'. The table has four columns: EMAIL, USER NAME, ROLL NUMBER, and PASSWORD. There are two rows of data. An 'Export to CSV' button is visible in the top right corner.

EMAIL	USER NAME	ROLL NUMBER	PASSWORD
123@gmail.com	kumar	2111	123@ibm
456@gmail.com	ram	2112	456@ibm

UPDATE:



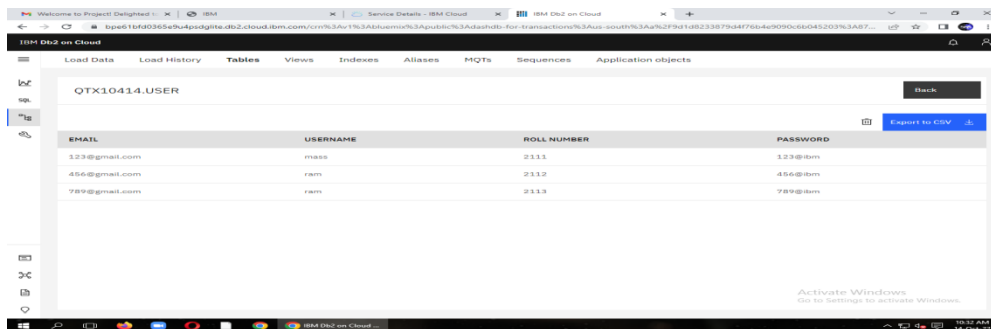
The screenshot shows the IBM Db2 on Cloud SQL Editor. A script is being executed to update the 'QTX10414.USER' table. The script contains the following SQL statements:

```
1 INSERT INTO USER VALUES ('123@gmail.com', 'kumar', '2111', '123@ibm');
2 INSERT INTO USER VALUES ('456@gmail.com', 'ram', '2112', '456@ibm');
3 INSERT INTO USER VALUES ('789@gmail.com', 'ram', '2113', '789@ibm');
4 UPDATE USER SET USERNAME='mass' WHERE EMAIL='123@gmail.com';
5 UPDATE USER SET USERNAME='mass' WHERE EMAIL='456@gmail.com';
6 UPDATE USER SET USERNAME='mass' WHERE EMAIL='789@gmail.com';
```

The 'History' tab shows the execution results of the script.

Script	Date	Status	Runtime
UNTESTED - 5	OCT 24, 2022 10:21:44 AM	Success	0.007 s
UPDATE USER SET USERNAME='mass' WHERE EMAIL='123@gmail.com'	OCT 24, 2022 10:21:47 AM	Success	0.012 s
UPDATE USER SET USERNAME='mass' WHERE EMAIL='456@gmail.com'	OCT 24, 2022 10:21:52 AM	Success	0.012 s
UPDATE USER SET USERNAME='mass' WHERE EMAIL='789@gmail.com'	OCT 24, 2022 10:21:57 AM	Success	0.007 s

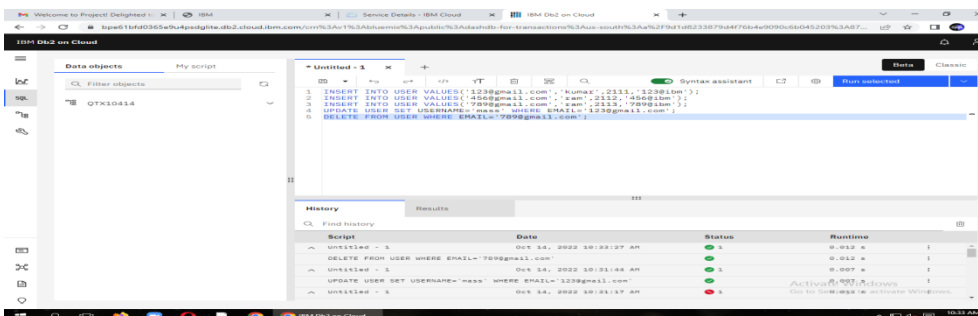
Table View:



The screenshot shows the IBM Db2 on Cloud interface. The 'Tables' tab is selected, displaying a table named 'QTX10414.USER'. The table has four columns: EMAIL, USERNAME, ROLL NUMBER, and PASSWORD. There are three rows of data. An 'Export to CSV' button is visible in the top right corner.

EMAIL	USERNAME	ROLL NUMBER	PASSWORD
123@gmail.com	mass	2111	123@ibm
456@gmail.com	ram	2112	456@ibm
789@gmail.com	ram	2113	789@ibm

DELETE:



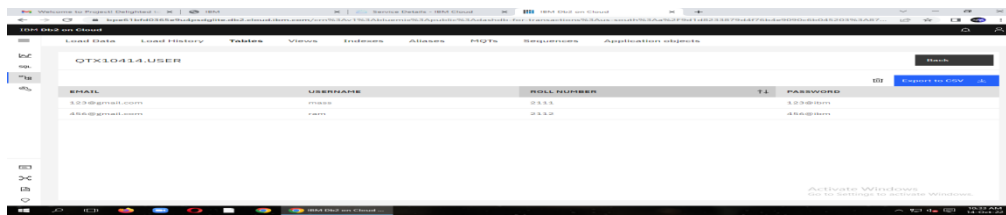
The screenshot shows the IBM Db2 on Cloud SQL Editor. A script is being executed to delete a row from the 'QTX10414.USER' table. The script contains the following SQL statements:

```
1 INSERT INTO USER VALUES ('123@gmail.com', 'kumar', '2111', '123@ibm');
2 INSERT INTO USER VALUES ('456@gmail.com', 'ram', '2112', '456@ibm');
3 INSERT INTO USER VALUES ('789@gmail.com', 'ram', '2113', '789@ibm');
4 UPDATE USER SET USERNAME='mass' WHERE EMAIL='123@gmail.com';
5 DELETE FROM USER WHERE EMAIL='789@gmail.com';
```

The 'History' tab shows the execution results of the script.

Script	Date	Status	Runtime
UNTESTED - 5	OCT 24, 2022 10:21:27 AM	Success	0.012 s
DELETE FROM USER WHERE EMAIL='789@gmail.com'	OCT 24, 2022 10:21:28 AM	Success	0.012 s
UNTESTED - 5	OCT 24, 2022 10:21:28 AM	Success	0.007 s
UPDATE USER SET USERNAME='mass' WHERE EMAIL='123@gmail.com'	OCT 24, 2022 10:21:37 AM	Success	0.012 s

TABLE View:



EMAIL	USERNAME	ROLL NUMBER	T1	PASSWORD
S.D.S@gmail.com	sdss	2020		S.D.S@sdss
A.B.S@gmail.com	abs	2020		A.B.S@sdss

3. Connect python with db2.

Solution:

```
import ibm_db
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=qtx10414;PWD=jMIXi3MGbJcaYc5E", "", "")
```

4. Create a flask app with the registration page. Login page and the 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 so the welcome page.

Solution:

app.py

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

```
import ibm_db
```

```
import bcrypt
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=qtx10414;PWD=jMIXi3MGbJcaYc5E", "", "")
```

```
# url_for('static', filename='style.css')
```

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

```
app.secret_key = b'_5#y2L"F4Q8z\xec]/'
```

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

```
def home():
```

```
    if 'email' not in session:
```

```
        return redirect(url_for('login'))
```

```
    return render_template('home.html', name='Home')
```

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

```
def register():
```

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

```
        email = request.form['email']
```

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

```

rollNo = request.form['rollNo']
password = request.form['password']

if not email or not username or not rollNo or not password:
    return render_template('register.html',error='Please fill all fields')

hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())

query = "SELECT * FROM USER WHERE email=? OR ROLLNO=?"
stmt = ibm_db.prepare(conn, query)
ibm_db.bind_param(stmt,1,email)
ibm_db.bind_param(stmt,2,rollNo)
ibm_db.execute(stmt)
isUser = ibm_db.fetch_assoc(stmt)

if not isUser:
    insert_sql = "INSERT INTO USER(EMAIL, USERNAME, PASSWORD, ROLLNO) VALUES (?, ?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, email)
    ibm_db.bind_param(prep_stmt, 2, username)
    ibm_db.bind_param(prep_stmt, 3, hash)
    ibm_db.bind_param(prep_stmt, 4, rollNo)
    ibm_db.execute(prep_stmt)
    return render_template('register.html',success="You can login")
else:
    return render_template('register.html',error='Invalid Credentials')

return render_template('register.html',name='Home')

@app.route("/login",methods=['GET','POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']

        if not email or not password:
            return render_template('login.html',error='Please fill all fields')
        query = "SELECT * FROM USER WHERE email=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        isUser = ibm_db.fetch_assoc(stmt)
        print(isUser,password)

        if not isUser:
            return render_template('login.html',error='Invalid Credentials')

        isPasswordMatch = bcrypt.checkpw(password.encode('utf-8'),isUser['PASSWORD'].encode('utf-8'))

        if not isPasswordMatch:
            return render_template('login.html',error='Invalid Credentials')

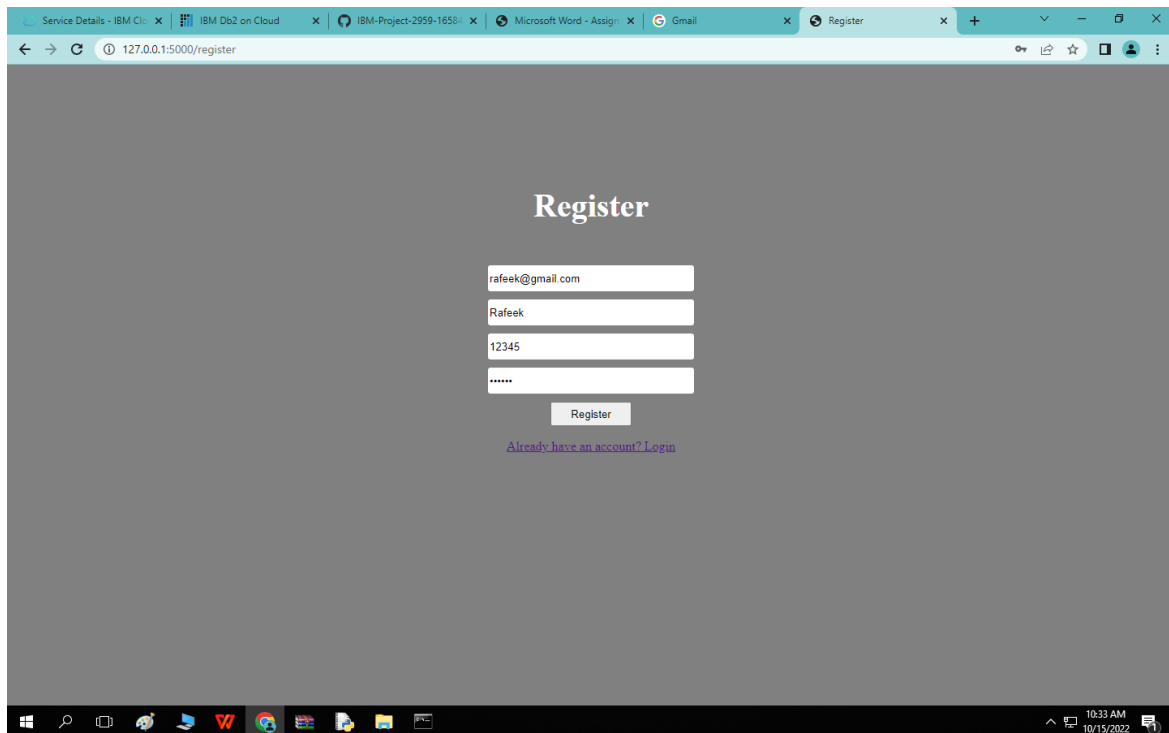
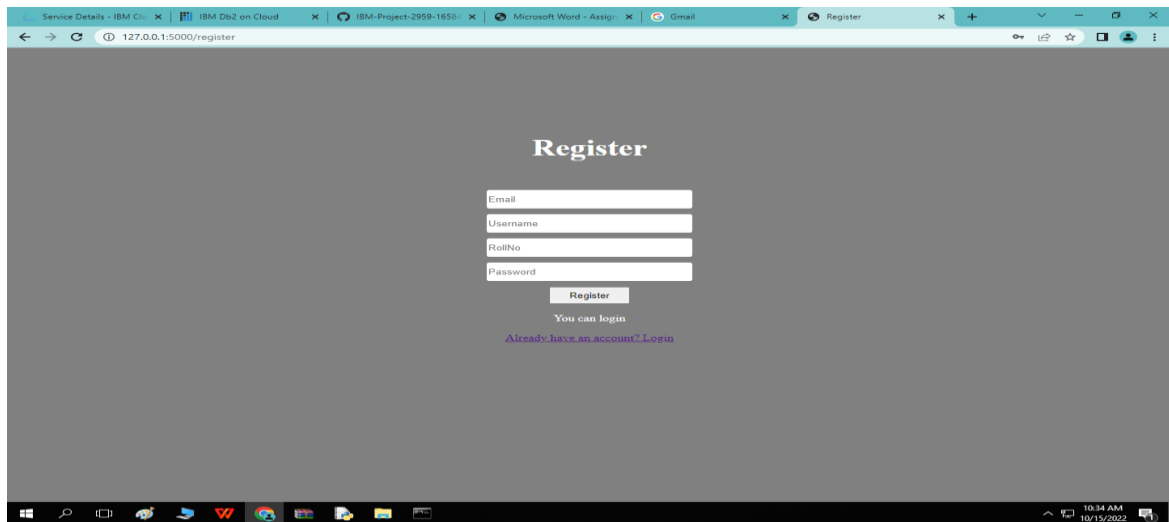
        session['email'] = isUser['EMAIL']
        return redirect(url_for('home'))

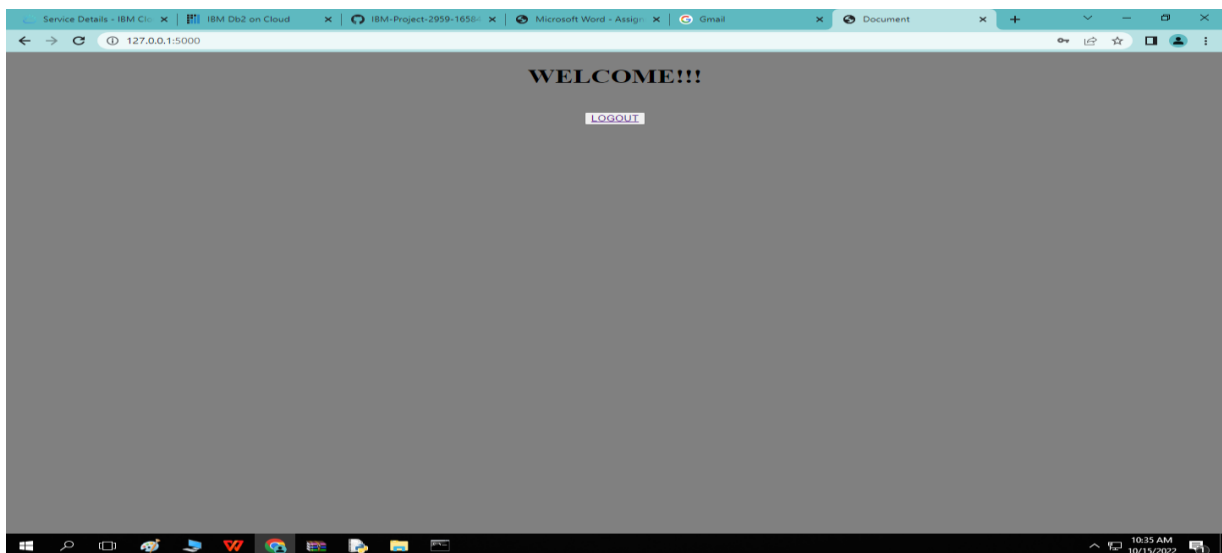
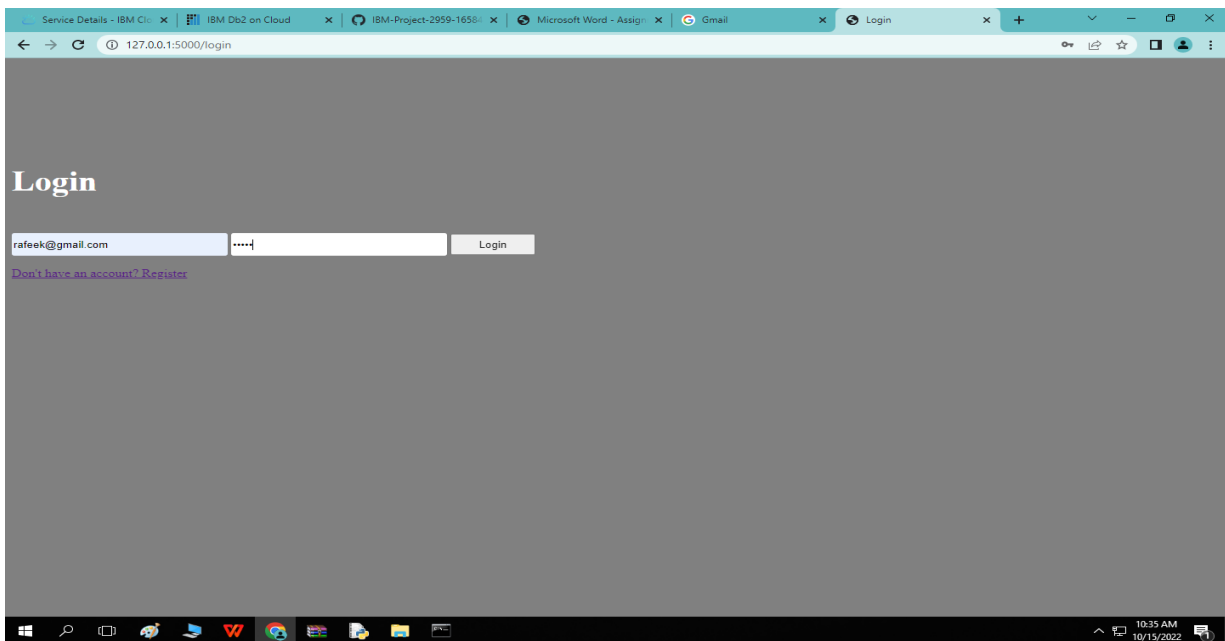
    return render_template('login.html',name='Home')

```

```
@app.route('/logout')
def logout():
    session.pop('email', None)
    return redirect(url_for('login'))
if __name__ == "__main__":
    app.run(debug=True)
```

OUTPUT:





Database:

