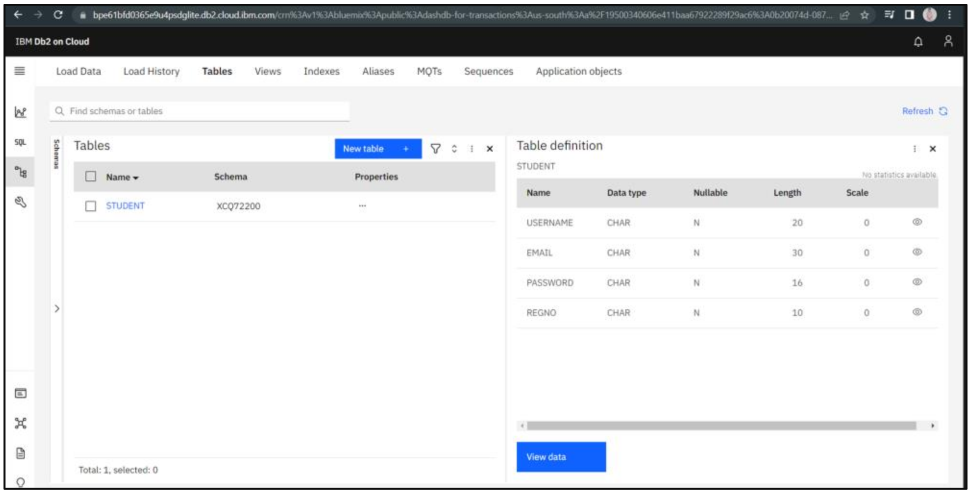
Assignment –2

DB2 Database Connection

Assignment Date	**24 September 2022**
Student Name	**Priyadharshini P**
Student Roll Number	**2019103562**
Maximum Marks	**2 Marks**

Questions:

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


bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F19500340606e411baa67922289f29ac6%3A0b20074d-087...
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
Find schemas or tables
Refresh
Schemas
New table +
View data

Tables

Name	Schema	Properties
STUDENT	XCQ72200	···

Total: 1, selected: 0

Table definition

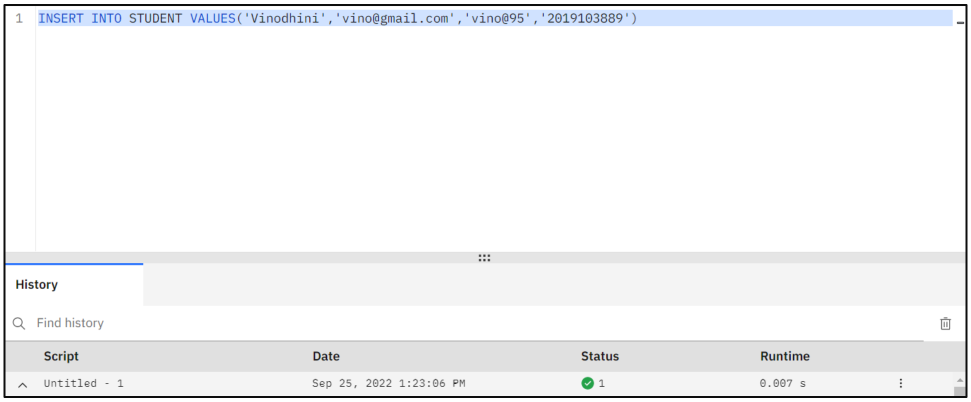
STUDENT

No statistics available

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	N	20	0
EMAIL	CHAR	N	30	0
PASSWORD	CHAR	N	16	0
REGNO	CHAR	N	10	0

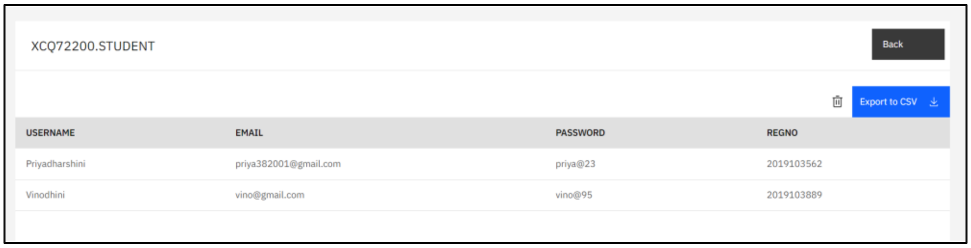
2. Perform UPDATE, DELETE queries with User table.

INSERT



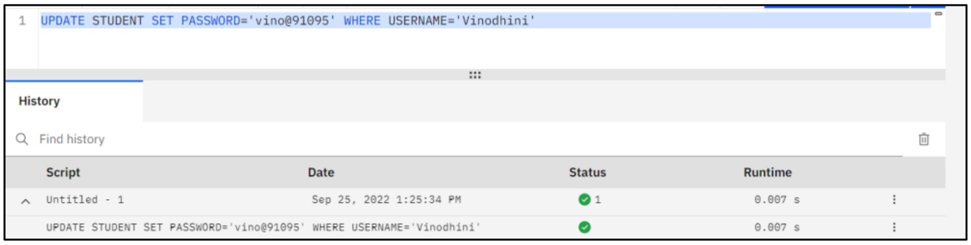
```
INSERT INTO STUDENT VALUES('Vinodhini','vino@gmail.com','vino@95','2019103889')
```

Script	Date	Status	Runtime
Untitled - 1	Sep 25, 2022 1:23:06 PM	✓ 1	0.007 s



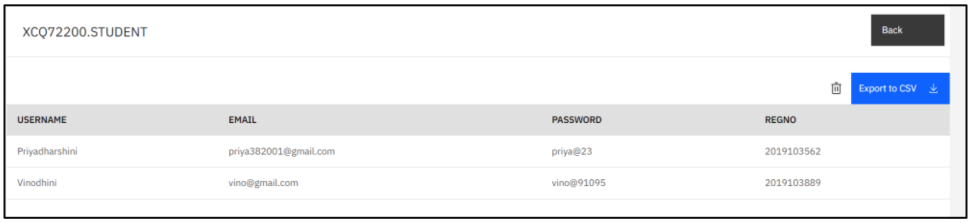
USERNAME	EMAIL	PASSWORD	REGNO
Priyadharshini	priya382001@gmail.com	priya@23	2019103562
Vinodhini	vino@gmail.com	vino@95	2019103889

UPDATE



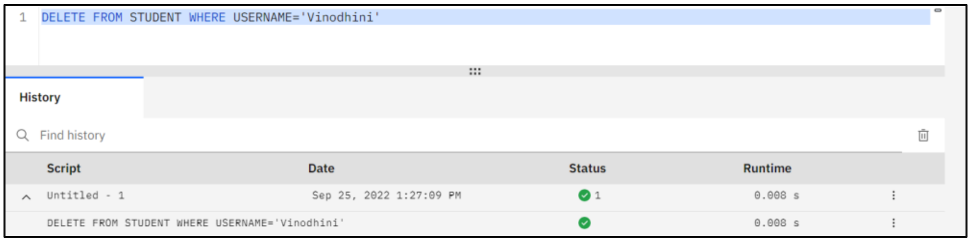
```
UPDATE STUDENT SET PASSWORD='vino@91095' WHERE USERNAME='Vinodhini'
```

Script	Date	Status	Runtime
Untitled - 1	Sep 25, 2022 1:25:34 PM	✓ 1	0.007 s
UPDATE STUDENT SET PASSWORD='vino@91095' WHERE USERNAME='Vinodhini'		✓	0.007 s



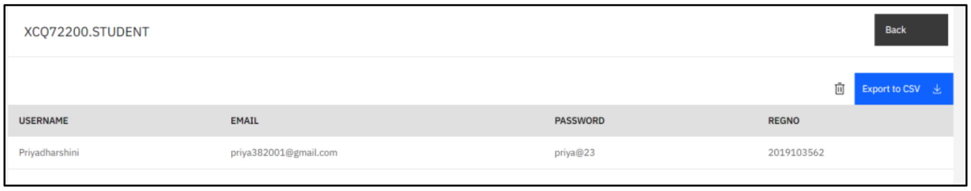
USERNAME	EMAIL	PASSWORD	REGNO
Priyadharshini	priya382001@gmail.com	priya@23	2019103562
Vinodhini	vino@gmail.com	vino@91095	2019103889

DELETE



```
DELETE FROM STUDENT WHERE USERNAME='Vinodhini'
```

Script	Date	Status	Runtime
Untitled - 1	Sep 25, 2022 1:27:09 PM	✓ 1	0.008 s
DELETE FROM STUDENT WHERE USERNAME='Vinodhini'		✓	0.008 s



USERNAME	EMAIL	PASSWORD	REGNO
Priyadharshini	priya382001@gmail.com	priya@23	2019103562

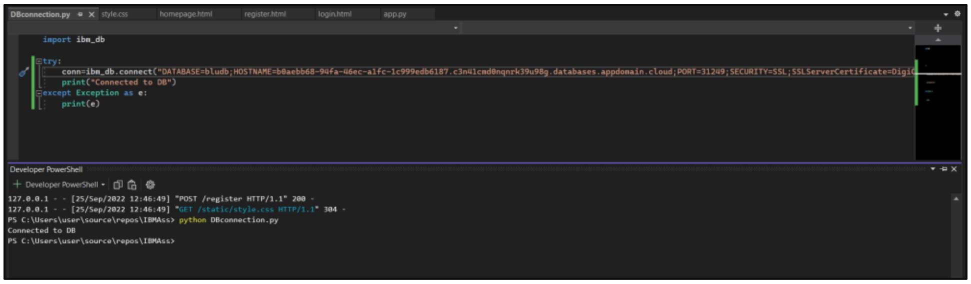
3. Connect python code to DB2.

Solution:

import ibm_db

try:

```
    conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=xcq72200;PWD=LMV8M41c4stsjNXi;", "", "")
    print("Connected to DB")
except Exception as e:
    print(e)
```



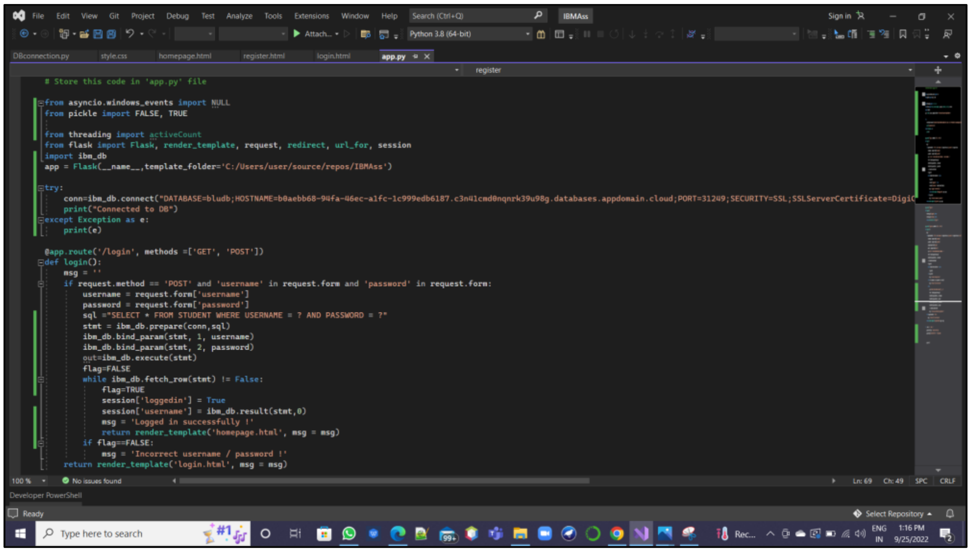
```
import ibm_db

try:
    conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;SSLServerCertificate=DigiC
    print("Connected to DB")
except Exception as e:
    print(e)
```

```
127.0.0.1 - - [25/Sep/2022 12:46:49] "POST /register HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2022 12:46:49] "GET /static/style.css HTTP/1.1" 304 -
PS C:\Users\user\source\repos\IBMAss> python DBconnection.py
Connected to DB
PS C:\Users\user\source\repos\IBMAss>
```

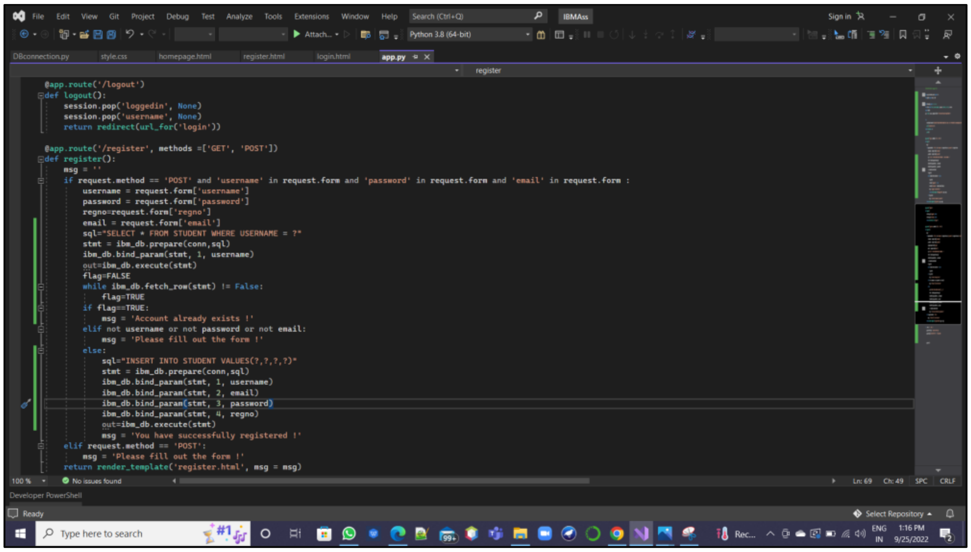
4. Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user with username and password. If the user is valid show the welcome page.

CODE

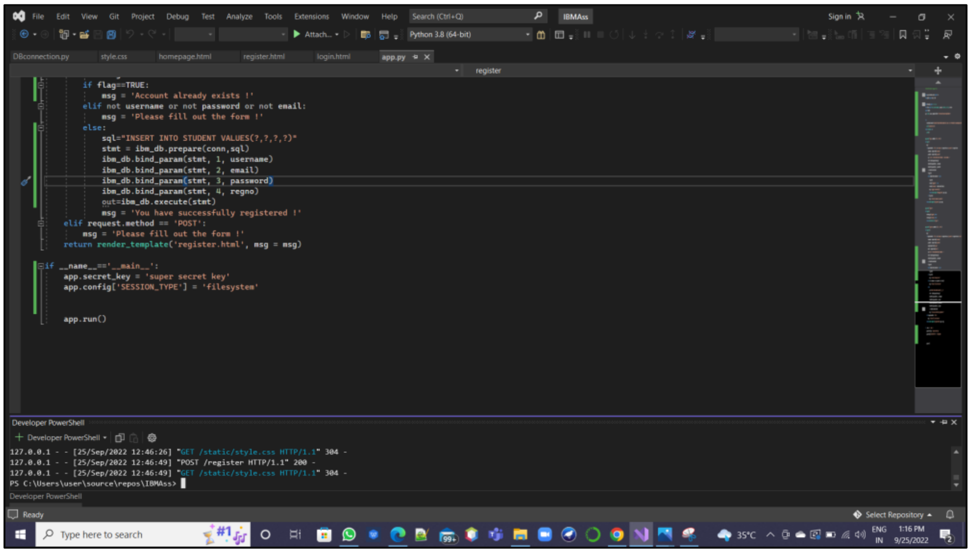
App.py



```
# Store this code in 'app.py' file

from asyncio.windows_events import NULL
from pickle import FALSE, TRUE

from threading import activeCount
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
app = Flask(__name__,template_folder='C:/Users/user/source/repos/IBMAss')

try:
    conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;SSLServerCertificate=DigiC
    print("Connected to DB")
except Exception as e:
    print(e)

@app.route('/login', methods =['GET', 'POST'])
def login():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        sql ="SELECT * FROM STUDENT WHERE USERNAME = ? AND PASSWORD = ?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        out=ibm_db.execute(stmt)
        flag=FALSE
        while ibm_db.fetch_row(stmt) != False:
            flag=TRUE
            session['loggedin'] = True
            session['username'] = ibm_db.result(stmt,0)
            msg = 'Logged in successfully !'
            return render_template('homepage.html', msg = msg)
        if flag==FALSE:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)
```



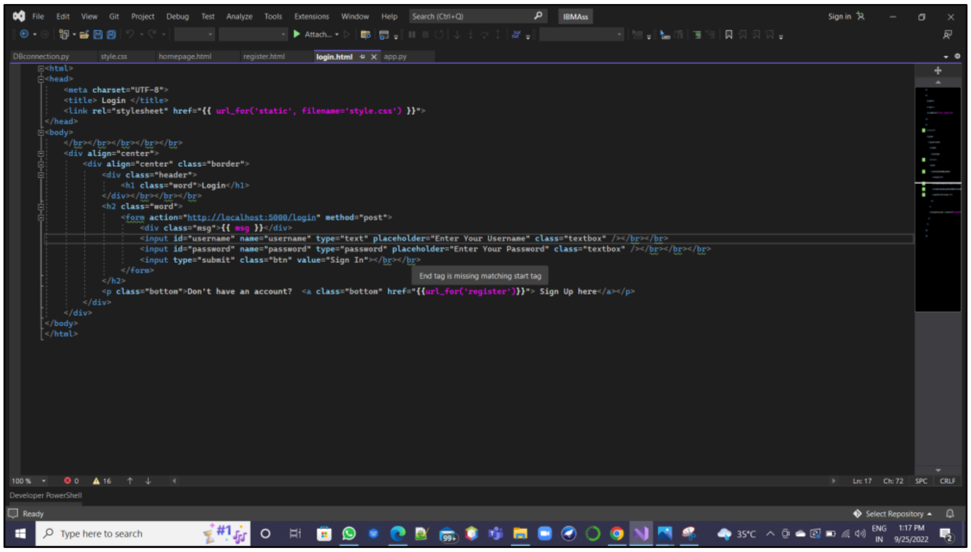
```
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('username', None)
    return redirect(url_for('login'))

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form :
        username = request.form['username']
        password = request.form['password']
        regno=request.form['regno']
        email = request.form['email']
        sql="SELECT * FROM STUDENT WHERE USERNAME = ?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt, 1, username)
        out=ibm_db.execute(stmt)
        flag=FALSE
        while ibm_db.fetch_row(stmt) != False:
            flag=TRUE
        if flag==TRUE:
            msg = 'Account already exists !'
        elif not username or not password or not email:
            msg = 'Please fill out the form !'
        else:
            sql="INSERT INTO STUDENT VALUES(?,?,?,?)"
            stmt = ibm_db.prepare(conn,sql)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email)
            ibm_db.bind_param(stmt, 3, password)
            ibm_db.bind_param(stmt, 4, regno)
            out=ibm_db.execute(stmt)
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)
```



```
        if flag==TRUE:
            msg = 'Account already exists !'
        elif not username or not password or not email:
            msg = 'Please fill out the form !'
        else:
            sql="INSERT INTO STUDENT VALUES(?,?,?,?)"
            stmt = ibm_db.prepare(conn,sql)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email)
            ibm_db.bind_param(stmt, 3, password)
            ibm_db.bind_param(stmt, 4, regno)
            out=ibm_db.execute(stmt)
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)

if __name__=='__main__':
    app.secret_key = 'super secret key'
    app.config['SESSION_TYPE'] = 'filesystem'


    app.run()
```

```
127.0.0.1 - - [25/Sep/2022 12:46:26] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Sep/2022 12:46:49] "POST /register HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2022 12:46:49] "GET /static/style.css HTTP/1.1" 304 -
PS C:\Users\user\source\repos\IBMAss>
```

Login.html



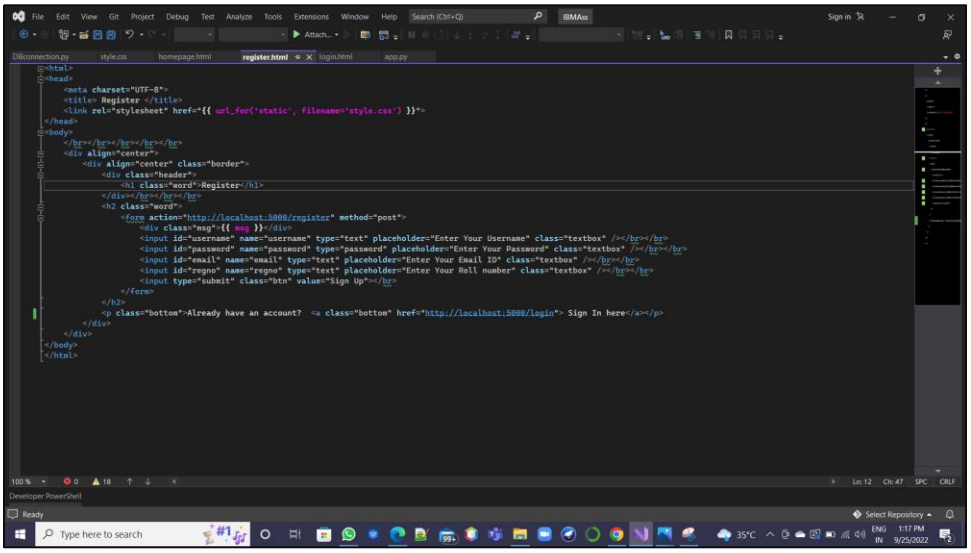
File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) IBMAss Sign in
Attach...
DBconnection.py style.css homepage.html register.html login.html app.py

```html
<html>
<head>
    <meta charset="UTF-8">
    <title> Login </title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    </br></br></br></br></br>
    <div align="center">
        <div align="center" class="border">
            <div class="header">
                <h1 class="word">Login</h1>
            </div></br></br></br>
            <h2 class="word">
                <form action="http://localhost:5000/login" method="post">
                    <div class="msg">{{ msg }}</div>
                    <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox" /></br></br>
                    <input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox" /></br></br></br>
                    <input type="submit" class="btn" value="Sign In"></br></br>
                </form>
            </h2>
            <p class="bottom">Don't have an account?  <a class="bottom" href="{{url_for('register')}}"> Sign Up here</a></p>
        </div>
    </div>
</body>
</html>
```

End tag is missing matching start tag

100 % 0 16 Ln: 17 Ch: 72 SPC CRLF
Developer PowerShell
Ready Select Repository
Type here to search
35°C ENG IN 1:17 PM 9/25/2022

Register.html

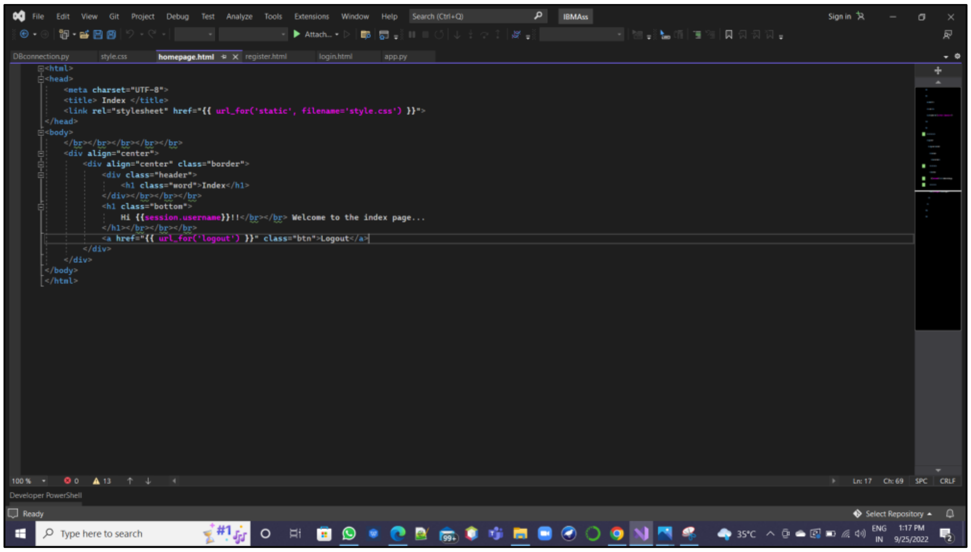


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) IBMAss Sign in
Attach...
DBconnection.py style.css homepage.html register.html login.html app.py

```html
<html>
<head>
    <meta charset="UTF-8">
    <title> Register </title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    </br></br></br></br></br>
    <div align="center">
        <div align="center" class="border">
            <div class="header">
                <h1 class="word">Register</h1>
            </div></br></br></br>
            <h2 class="word">
                <form action="http://localhost:5000/register" method="post">
                    <div class="msg">{{ msg }}</div>
                    <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox" /></br></br>
                    <input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox" /></br></br>
                    <input id="email" name="email" type="text" placeholder="Enter Your Email ID" class="textbox" /></br></br>
                    <input id="regno" name="regno" type="text" placeholder="Enter Your Roll number" class="textbox" /></br></br>
                    <input type="submit" class="btn" value="Sign Up"></br>
                </form>
            </h2>
            <p class="bottom">Already have an account?  <a class="bottom" href="http://localhost:5000/login"> Sign In here</a></p>
        </div>
    </div>
</body>
</html>
```

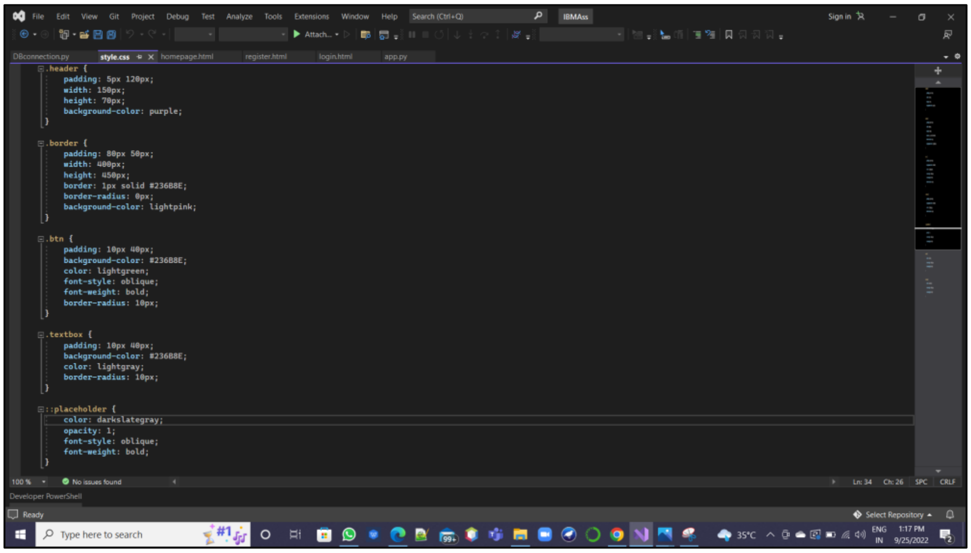
100 % 0 18 Ln: 12 Ch: 47 SPC CRLF
Developer PowerShell
Ready Select Repository
Type here to search
35°C ENG IN 1:17 PM 9/25/2022

Homepage.html

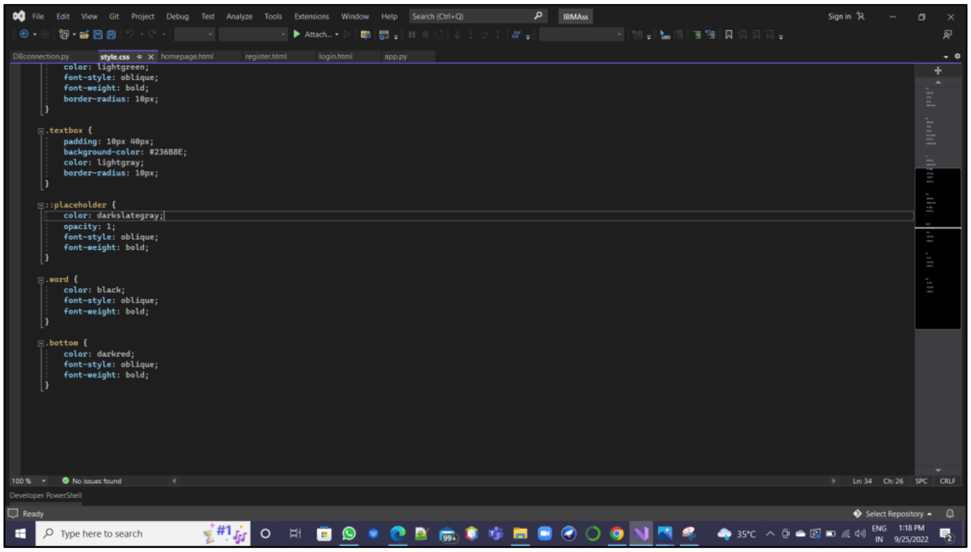


```html
<html>
<head>
    <meta charset="UTF-8">
    <title> Index </title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    </br></br></br></br></br>
    <div align="center">
        <div align="center" class="border">
            <div class="header">
                <h1 class="word">Index</h1>
            </div></br></br></br>
            <h1 class="bottom">
                Hi {{session.username}}!!</br></br> Welcome to the index page...
            </h1></br></br></br>
            <a href="{{ url_for('logout') }}" class="btn">Logout</a>
        </div>
    </div>
</body>
</html>
```

Style.css



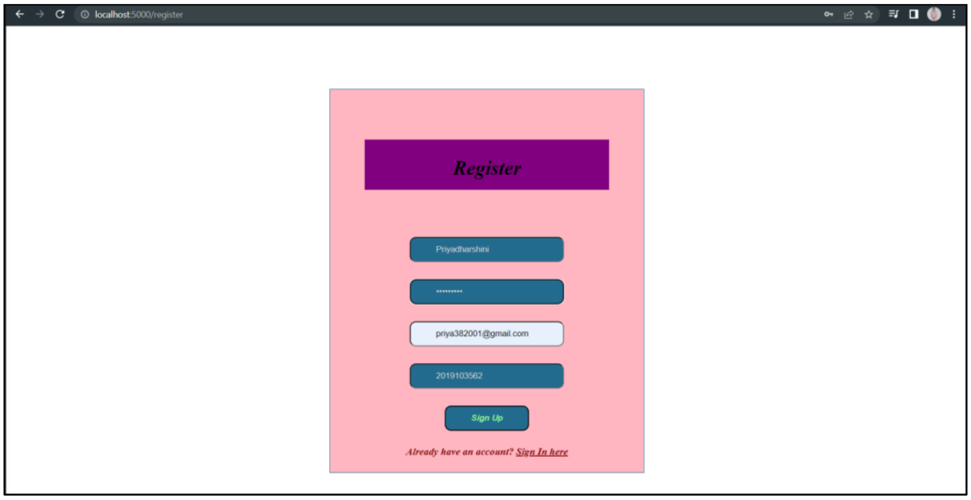
```css
.header {
    padding: 5px 120px;
    width: 150px;
    height: 70px;
    background-color: purple;
}

.border {
    padding: 80px 50px;
    width: 400px;
    height: 450px;
    border: 1px solid #236B8E;
    border-radius: 0px;
    background-color: lightpink;
}

.btn {
    padding: 10px 40px;
    background-color: #236B8E;
    color: lightgreen;
    font-style: oblique;
    font-weight: bold;
    border-radius: 10px;
}

.textbox {
    padding: 10px 40px;
    background-color: #236B8E;
    color: lightgray;
    border-radius: 10px;
}

::placeholder {
    color: darkslategray;
    opacity: 1;
    font-style: oblique;
    font-weight: bold;
}
```



```
        color: lightgreen;
        font-style: oblique;
        font-weight: bold;
        border-radius: 10px;
}

.textbox {
        padding: 10px 40px;
        background-color: #236B8E;
        color: lightgray;
        border-radius: 10px;
}

::placeholder {
        color: darkslategray;
        opacity: 1;
        font-style: oblique;
        font-weight: bold;
}

.word {
        color: black;
        font-style: oblique;
        font-weight: bold;
}

.bottom {
        color: darkred;
        font-style: oblique;
        font-weight: bold;
}
```

OUTPUT

REGISTER



Register

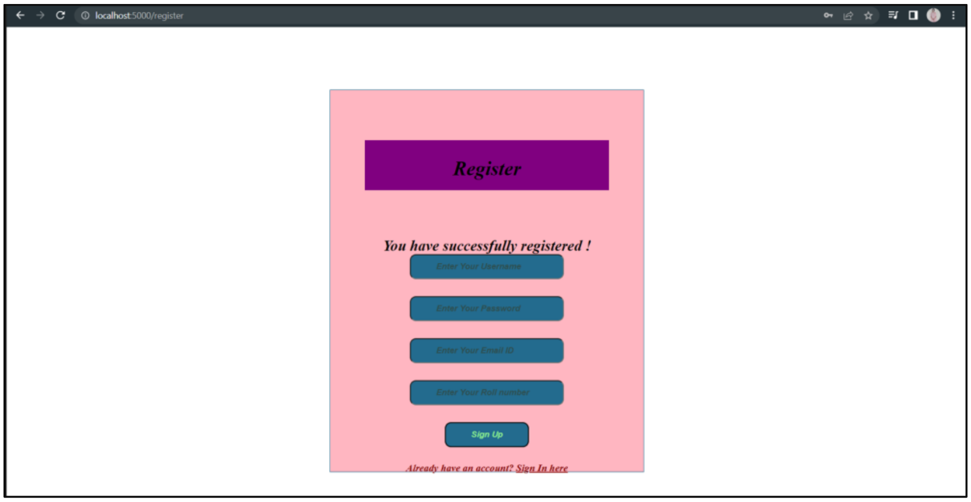
Priyadharshini

••••••••••

priya382001@gmail.com

2019103562

Already have an account? Sign In here



Register

You have successfully registered !

Enter Your Username

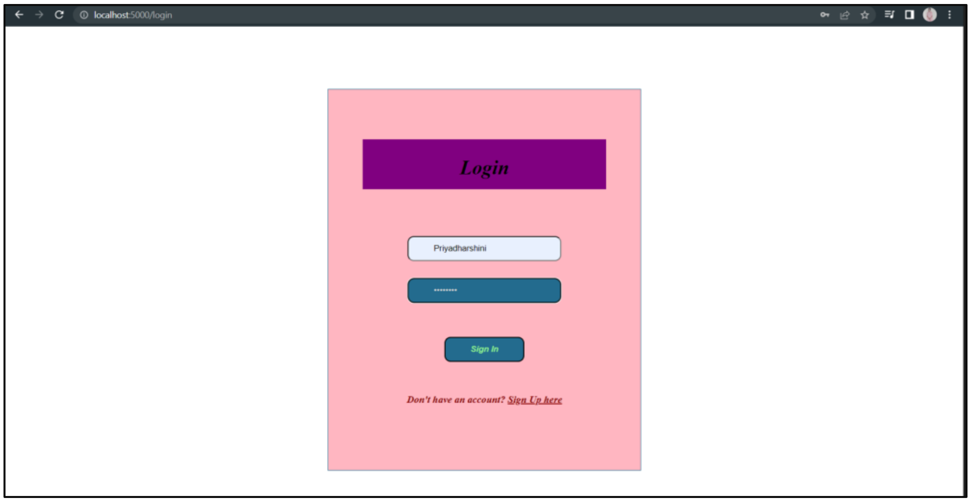
Enter Your Password

Enter Your Email ID

Enter Your Roll number

Already have an account? Sign In here

LOGIN



localhost:5000/login

Login

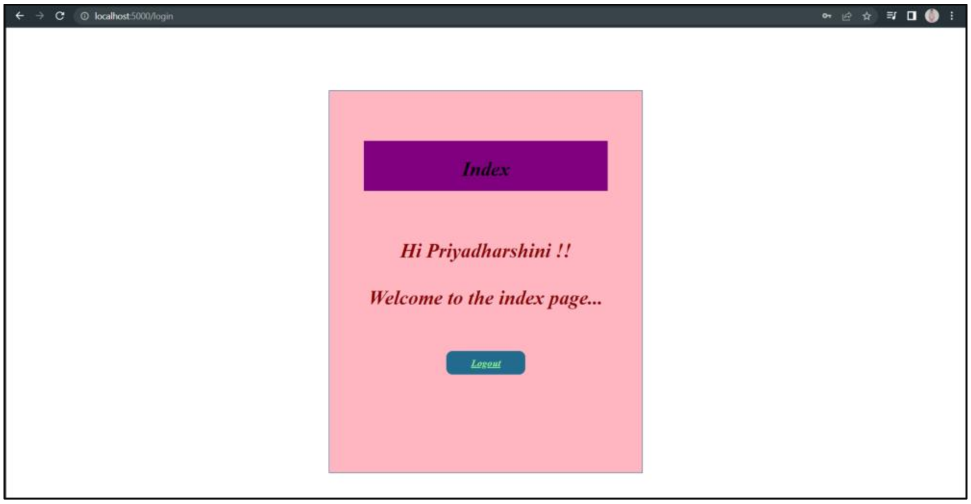
Priyadharshini

••••••••

Sign In

Don't have an account? Sign Up here

HOMEPAGE



localhost:5000/login

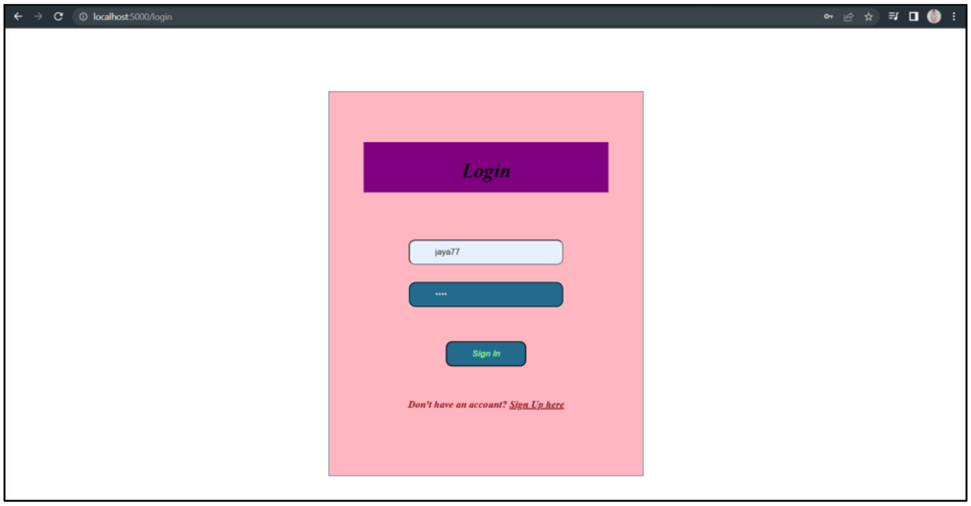
Index

Hi Priyadharshini !!

Welcome to the index page...

Logout

INCORRECT LOGIN



localhost:5000/login

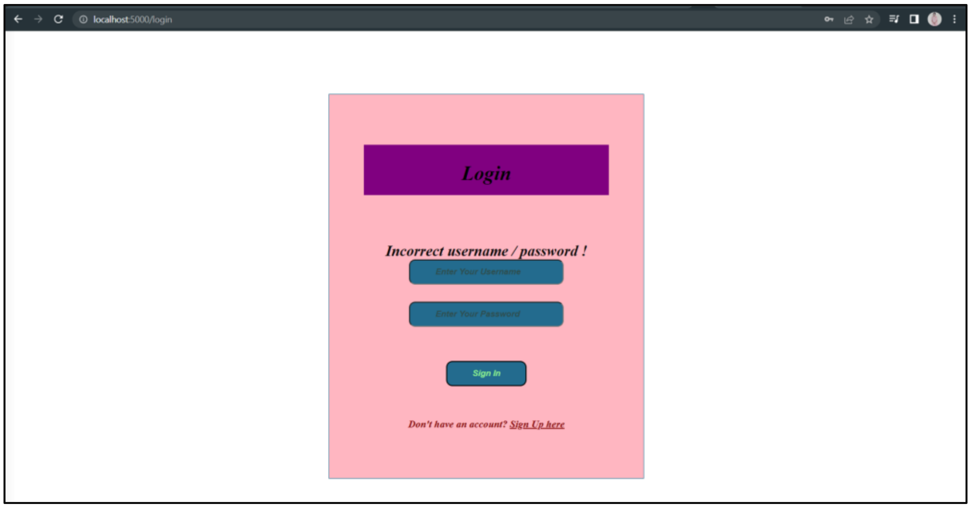
Login

jaya77

••••

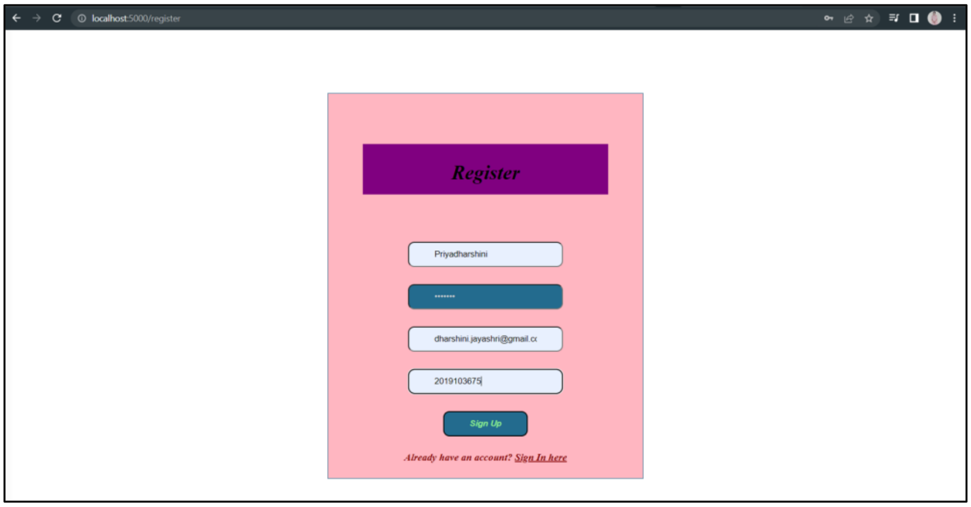
Sign In

Don't have an account? Sign Up here



Incorrect username / password !

USERNAME ALREADY EXISTS

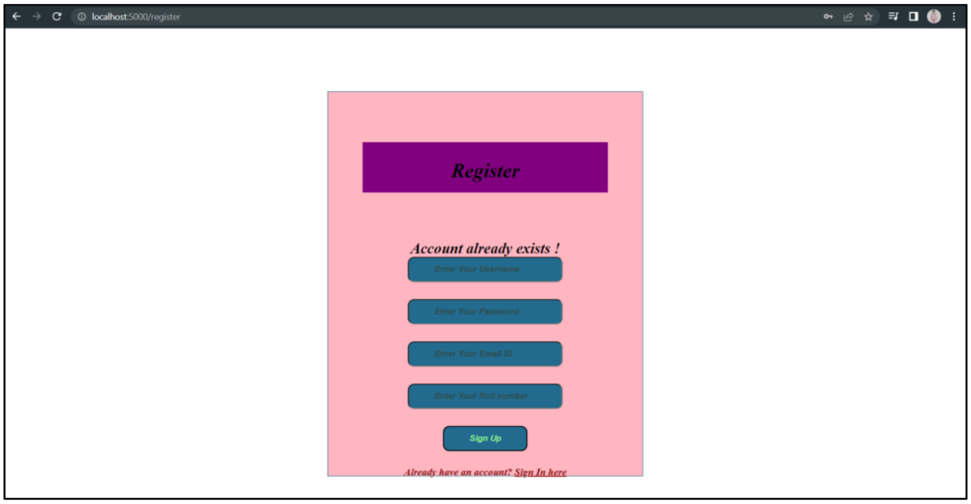


Priyadharshini

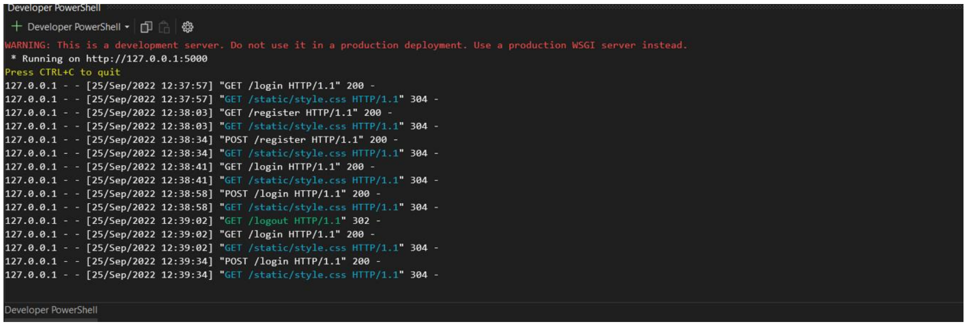
•••••••

dharshini.jayashri@gmail.cc

2019103675

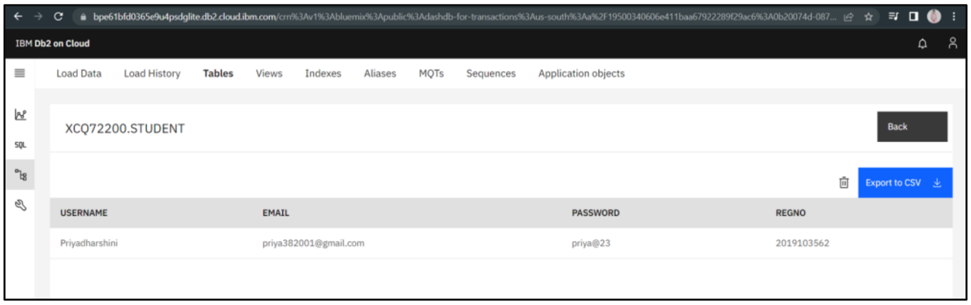


Account already exists !



```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [25/Sep/2022 12:37:57] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2022 12:37:57] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Sep/2022 12:38:03] "GET /register HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2022 12:38:03] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Sep/2022 12:38:34] "POST /register HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2022 12:38:34] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Sep/2022 12:38:41] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2022 12:38:41] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Sep/2022 12:38:58] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2022 12:38:58] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Sep/2022 12:39:02] "GET /logout HTTP/1.1" 302 -
127.0.0.1 - - [25/Sep/2022 12:39:02] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2022 12:39:02] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Sep/2022 12:39:34] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2022 12:39:34] "GET /static/style.css HTTP/1.1" 304 -
```

DATABASE TABLE



bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F19500340606e411baa67922289f29ac6%3A0b20074d-087...
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
SQL
Back
Export to CSV

XCQ72200.STUDENT

USERNAME	EMAIL	PASSWORD	REGNO
Priyadharshini	priya382001@gmail.com	priya@23	2019103562