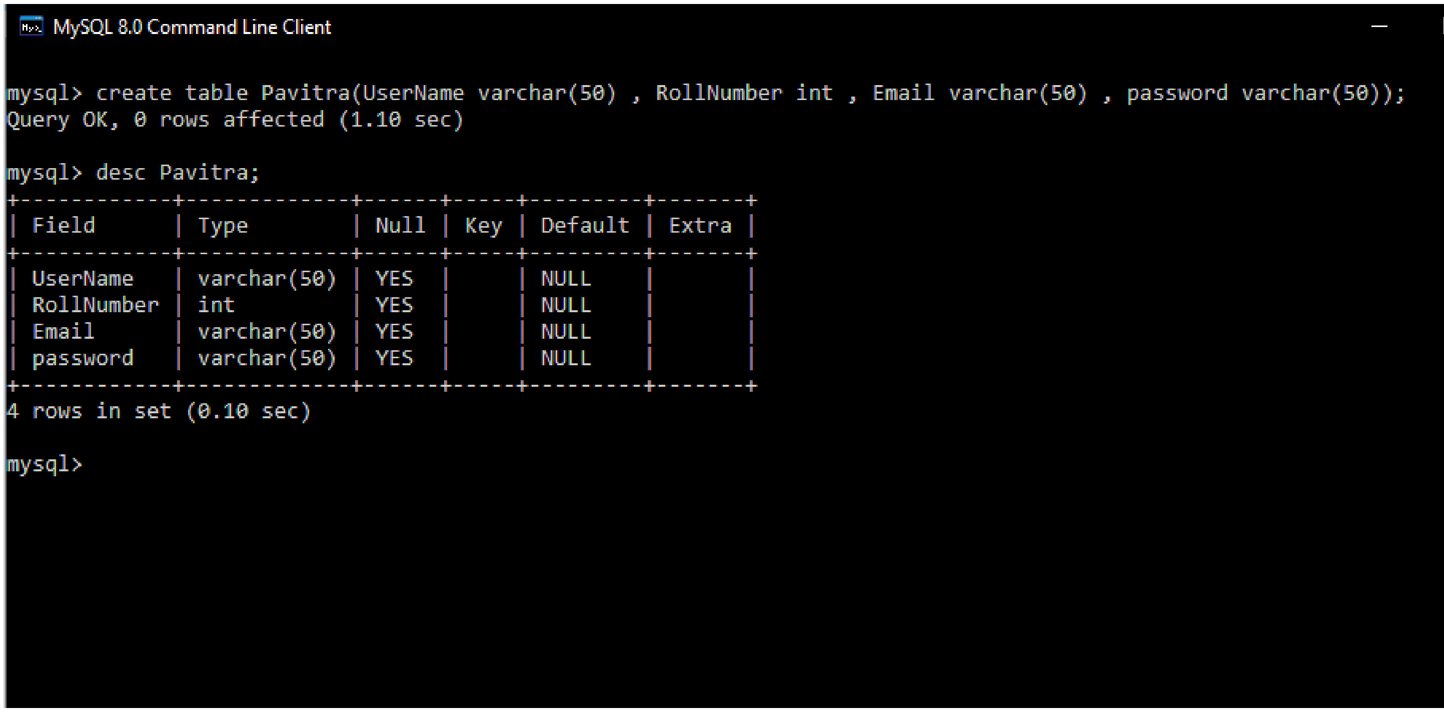
Date	23.10.22
Team id	PNT2022TMID49588
Project name	Personal expense tracker
Team member name	M.Pavithra

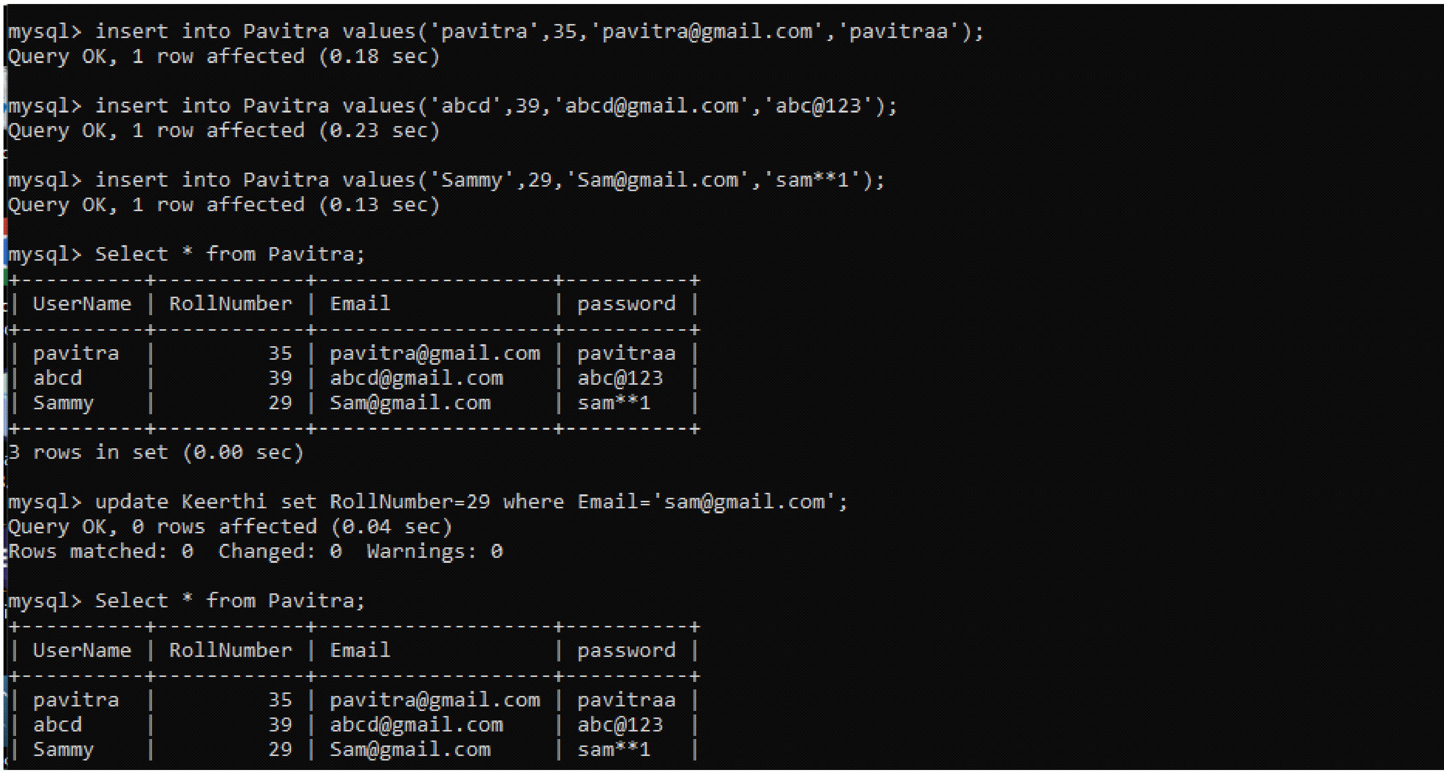
Assignment-2

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

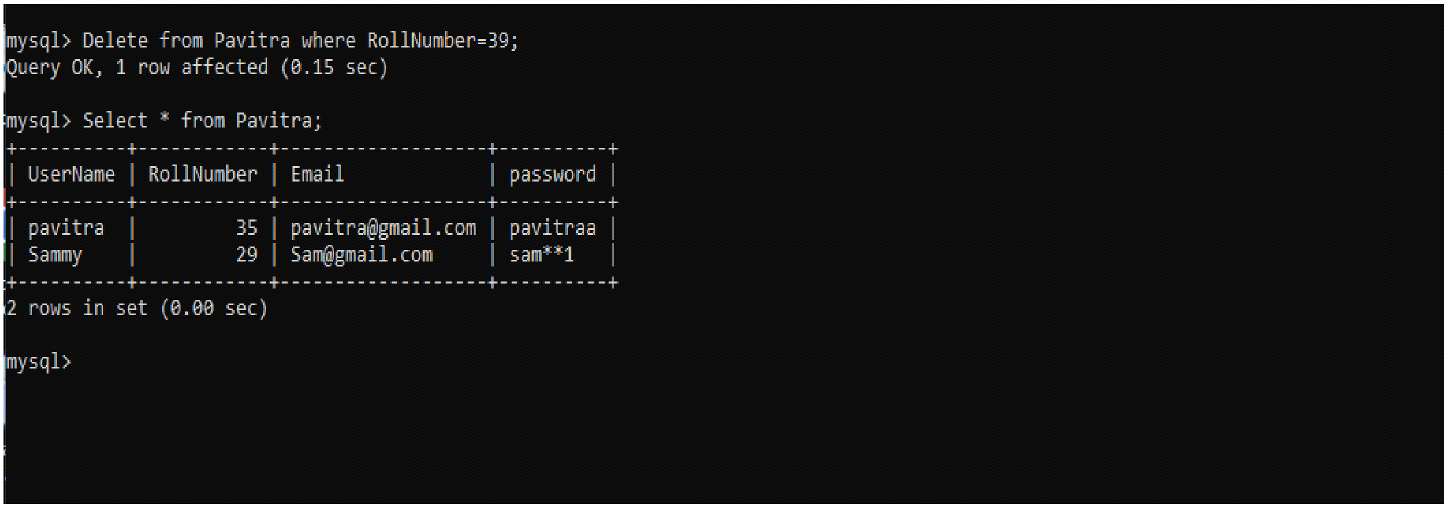


```
mysql> create table Pavitra(UserName varchar(50) , RollNumber int , Email varchar(50) , password varchar(50));
Query OK, 0 rows affected (1.10 sec)

mysql> desc Pavitra;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| UserName   | varchar(50) | YES  |     | NULL    |       |
| RollNumber | int         | YES  |     | NULL    |       |
| Email      | varchar(50) | YES  |     | NULL    |       |
| password   | varchar(50) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.10 sec)

mysql>
```

2.Perform UPDATE, DELETE Queries with user table.



```
mysql> insert into Pavitra values('pavitra',35,'pavitra@gmail.com','pavitraa');
Query OK, 1 row affected (0.18 sec)

mysql> insert into Pavitra values('abcd',39,'abcd@gmail.com','abc@123');
Query OK, 1 row affected (0.23 sec)

mysql> insert into Pavitra values('Sammy',29,'Sam@gmail.com','sam**1');
Query OK, 1 row affected (0.13 sec)

mysql> Select * from Pavitra;
+----------+------------+-------------------+----------+
| UserName | RollNumber | Email             | password |
+----------+------------+-------------------+----------+
| pavitra  |         35 | pavitra@gmail.com | pavitraa |
| abcd     |         39 | abcd@gmail.com    | abc@123  |
| Sammy    |         29 | Sam@gmail.com     | sam**1   |
+----------+------------+-------------------+----------+
3 rows in set (0.00 sec)

mysql> update Keerthi set RollNumber=29 where Email='sam@gmail.com';
Query OK, 0 rows affected (0.04 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql> Select * from Pavitra;
+----------+------------+-------------------+----------+
| UserName | RollNumber | Email             | password |
+----------+------------+-------------------+----------+
| pavitra  |         35 | pavitra@gmail.com | pavitraa |
| abcd     |         39 | abcd@gmail.com    | abc@123  |
| Sammy    |         29 | Sam@gmail.com     | sam**1   |
```



```
mysql> Delete from Pavitra where RollNumber=39;
Query OK, 1 row affected (0.15 sec)

mysql> Select * from Pavitra;
+----------+------------+-------------------+----------+
| UserName | RollNumber | Email             | password |
+----------+------------+-------------------+----------+
| pavitra  |         35 | pavitra@gmail.com | pavitraa |
| Sammy    |         29 | Sam@gmail.com     | sam**1   |
+----------+------------+-------------------+----------+
2 rows in set (0.00 sec)

mysql>
```

3.Connect python code to db2.

Conn = ibm_db.connect(“DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=tzk44983;PWD=8a9aQs4FR88yeEXC”,’’,’’)

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 the database and navigate to the login page to authenticate user username and password. If the user is valid show the welcome page

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



app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=tzk44983;PWD=8a9aQs4FR88yeEXC",",")
@app.route('/')
def home():
    return render_template('register.html')
@app.route('/login', methods =['GET', 'POST'])
def login():
    global userid
    msg = "
    if request.method == 'POST'and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        stmt = ibm_db.prepare(conn,'SELECT * FROM accounts WHERE username = ? AND password = ?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            session['loggedin'] = True
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
            return render_template('index.html', msg = msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)


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


@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
```

```
    username = request.form['username']
    password = request.form['password']
    email = request.form['email']
    sql = "SELECT * FROM accounts 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 characters and numbers !'
    elif not username or not password or not email:
        msg = 'Please fill out the form !'
    else:
        insert_sql = "INSERT INTO accounts VALUES (?, ?, ?)"
        stmt = ibm_db.prepare(conn,insert_sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.bind_param(stmt, 3, email)
        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.run(debug = True)
```

index.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>


<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="{{ url_for('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 type="submit" class="btn" value="Sign Up"></br>
        </form>
      </h2>
      <p class="bottom">Already have an account?  <a class="bottom"
href="{{url_for('login')}}"> Sign In here</a></p>
     </div>
    </div>
  </body>
</html>
```

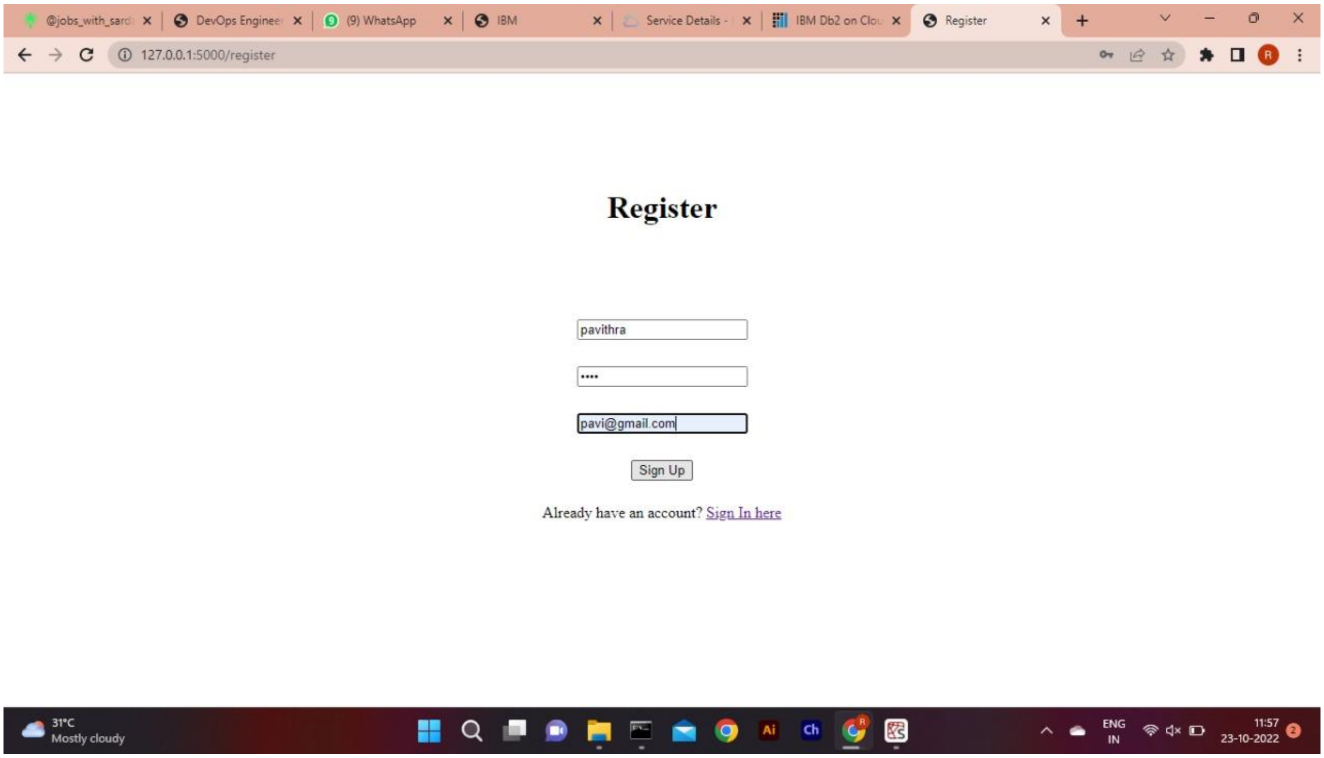
Style.css:

```
.header{
        padding: 5px 120px;
        width: 150px;
        height: 70px;
        background-color: #236B8E;
      }

      .border{
        padding: 80px 50px;
        width: 400px;
        height: 450px;
        border: 1px solid #236B8E;
        border-radius: 0px;
        background-color: #9AC0CD;
      }
```

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

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

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

.word{
   color: #FFFFFF;
```

```
    font-style: oblique;
    font-weight: bold;
}

.bottom{
    color: #236B8E;
    font-style: oblique;
    font-weight: bold;
}
```

Output:



Register

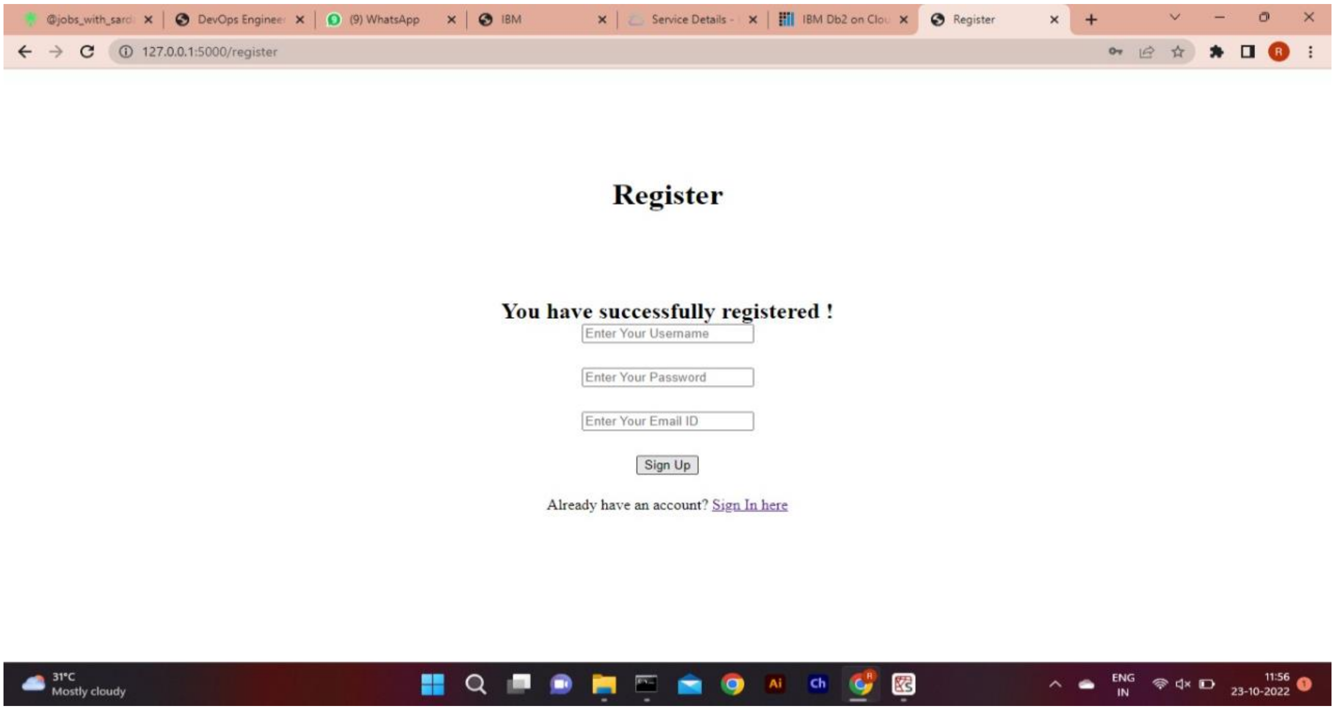
pavithra

••••

pavi@gmail.com

Sign Up

Already have an account? Sign In here



Register

You have successfully registered !

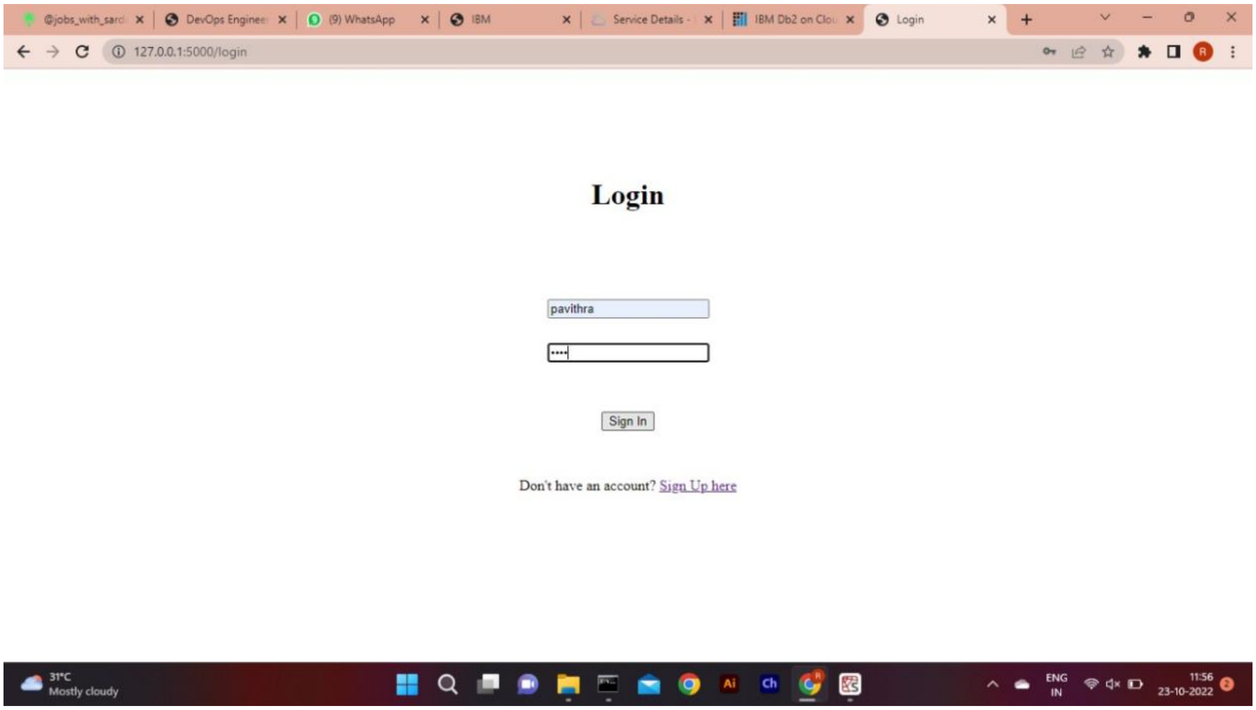
Enter Your Username

Enter Your Password

Enter Your Email ID

Sign Up

Already have an account? Sign In here

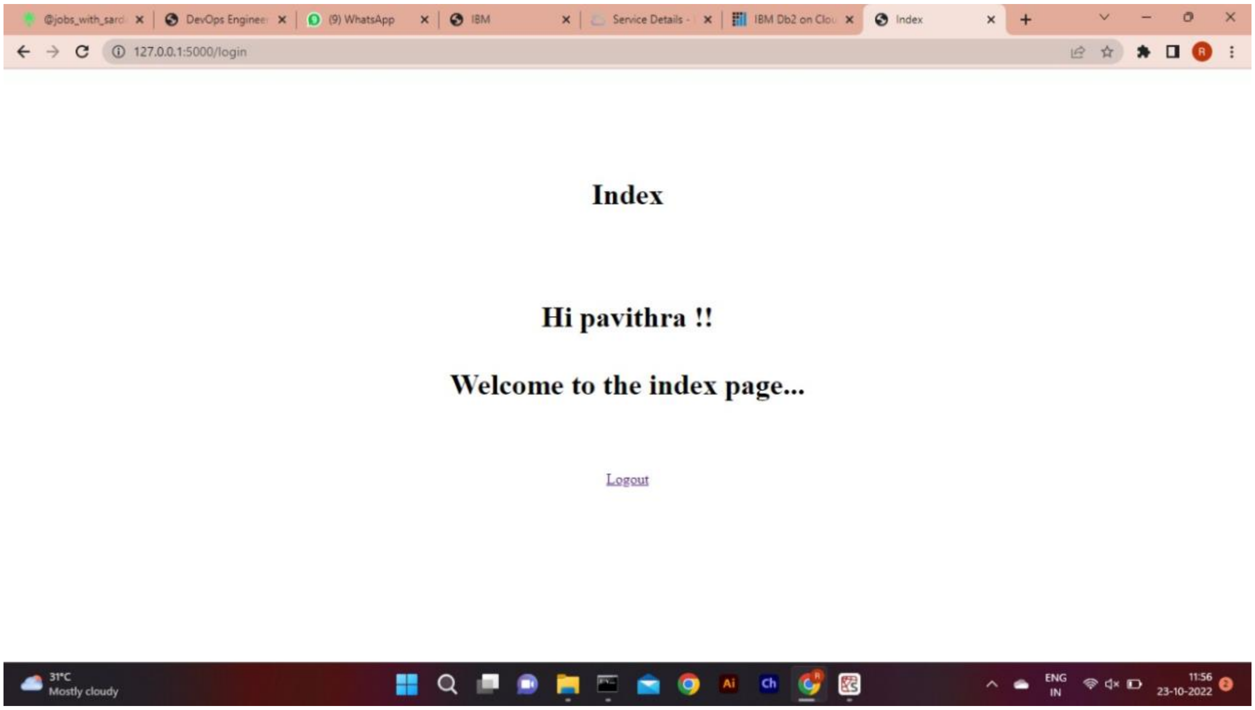


Login

pavithra

••••

Don't have an account? Sign Up here



Index

Hi pavithra !!

Welcome to the index page...

Logout