

PROJECT DEVELOPMENT PHASE

SPRINT-1

Team Id	PNT2022TMID44588
Project Title	Skill/Job Recommender Application
Date	17-11-2022

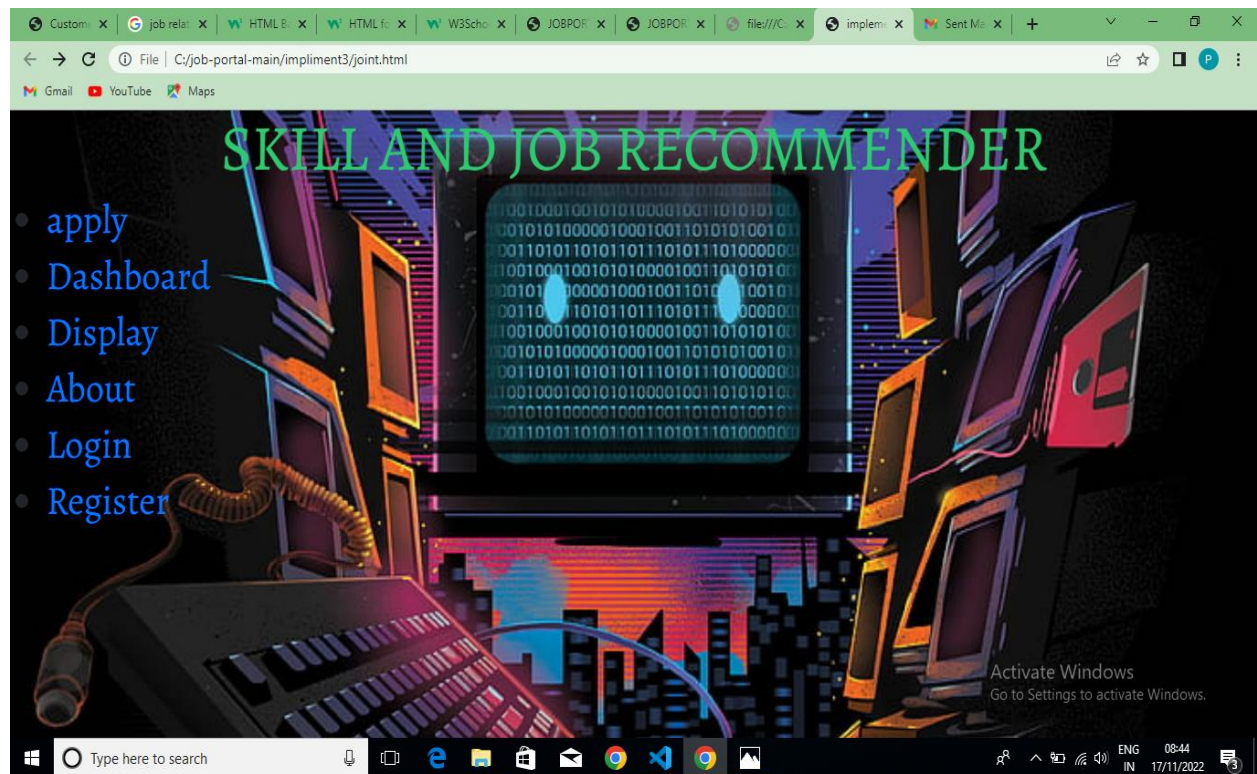
Home.html

```
<html>
  <head>
    <title>implement3</title>
    <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>JOBPORTAL | HOME</title>
    <meta charset="UTF-8">
    <!-- favicon -->
    <!-- <link rel="shortcut icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->
    <!-- <link rel="icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->
    <link rel="icon" type="image/png" sizes="16x16" href="/assets/img/favicon-32x32.png">
    <!-- 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://cdn.jsdelivr.net/npm/font-awesome/4.7.0/css/font-awesome.css">
    <!-- css stylesheet -->
    <link rel="stylesheet" href="{{url_for('static',filename='css/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>
  <center>
    <h1 style="font-size: 4rem; color: #2ecc71">SKILL AND JOB RECOMMENDER</h1>
  </center>
  <link rel="stylesheet" href="style.css">
  <center>
    <style>
      body {
        background-image: url('img4.jpg');
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-size: cover;
      }
    </style>
  </center>
</html>
```

```

        </style>
    </center>
</head><header id="header">
<nav>
    <ul>
        <h1><li><a href="apply.html">apply</a></li></h1>
        <h1><li><a href="dashboard.html">Dashboard</a></li></h1>
        <h1><li><a href="display.html">Display</a></li></h1>
        <h1><li><a href="about.html">About</a></li></h1>
        <h1><li><a href="login.html">Login</a></li></h1>
        <h1><li><a href="register.html">Register</a></li></h1>
    </ul>
</nav>
</header>
</html>

```



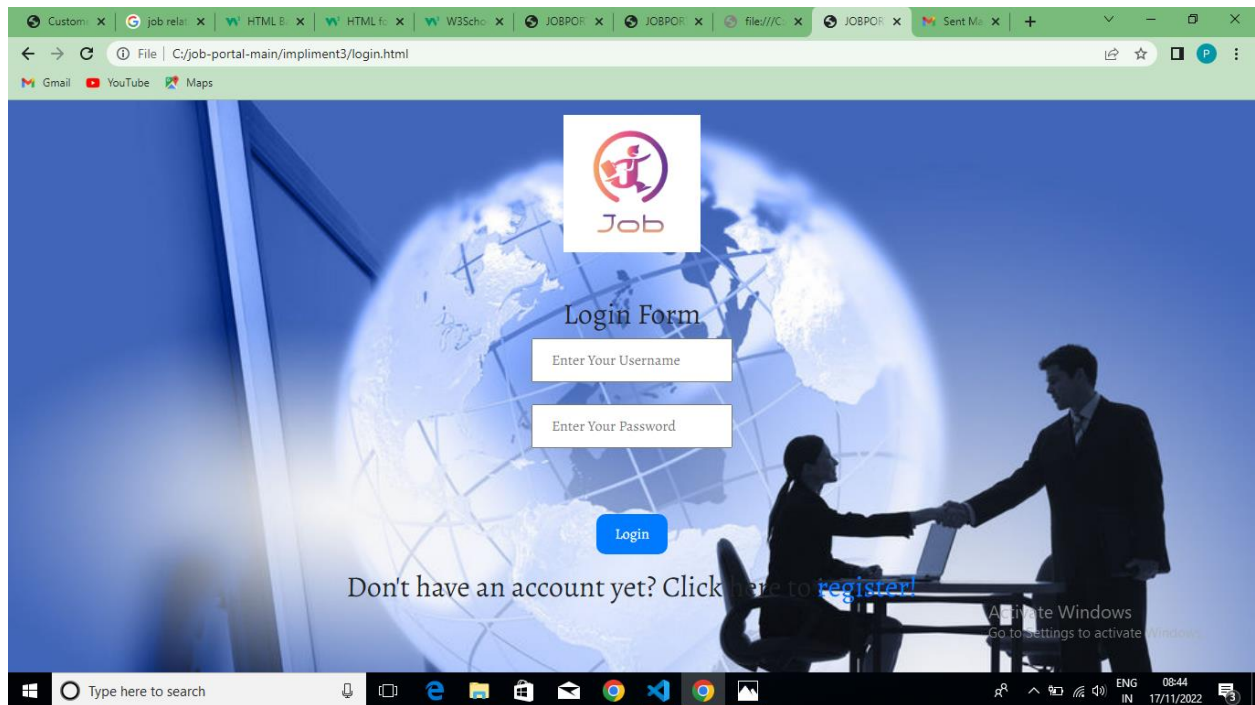
Login.html

```
<!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>JOBPORTAL | LOGIN</title>
  <link rel="stylesheet" href="style.css">
  <!-- favicon -->
  <!-- <link rel="shortcut icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->
  <!-- <link rel="icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->
  <link rel="icon" type="image/png" sizes="16x16" href="/assets/img/favicon-32x32.png">
  <!-- 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="{{url_for('static',filename='css/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>
<style>
  body {
    background-image: url('img12.jpg');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
  }
</style>
<body>
  <!-- bootstrap navbar -->
  <div class="logo mt-3 text-center">
    <a class="main-logo-img mt-5" href="#">
      <!-- <a class="navbar-brand" href="index.html">JobPortal</a> -->
    </a>
  </div>
  <!-- navbar ends -->
  <!-- Login form -->
  <div class="login text-center mt-5">
    <h2> Login Form </h2>
    <form action="/login" method="post">
      <input type="text" name="username" placeholder="Enter Your Username"
id="username" required></br></br>
```

```

        <input type="password" name="password" placeholder="Enter Your Password"
id="password" required></br></br>
        </br>
        </br>
        <button type="submit" id="button" class="btn btn-primary"> Login </button>
    </form>
</div>
<div class="note mt-3 text-center"> <!--Register form -->
<h2> Don't have an account yet? Click here to <a href="register.html">register! </a> </h2>
</div>
</body>
</html>

```



Registration page

```
<!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>JOBPORTAL | LOGIN</title>
  <link rel="stylesheet" href="style.css">

  <!-- favicon -->
  <!-- <link rel="shortcut icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->
  <!-- <link rel="icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->
  <link rel="icon" type="image/png" sizes="16x16" href="/assets/img/favicon-32x32.png">
  <!-- 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="{{url_for('static',filename='css/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>
<style>
  body {
    background-image: url('img11.jpg');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
  }
</style>
<body>
  <!-- bootstrap navbar -->
  <div class="logo mt-3 text-center">
    <a class="main-logo-img mt-5" href="#">
      <!-- <a class="navbar-brand" href="index.html">JobPortal</a> -->
    </a>
  </div>
  <!-- navbar ends -->
  <!-- Login form -->
  <div class="login text-center mt-5">
    <h2> Register Form </h2>
    <form action="/register" method="post">
      <!-- <input type="text" placeholder="fullname" id="fullname"> </br></br> -->
```

```

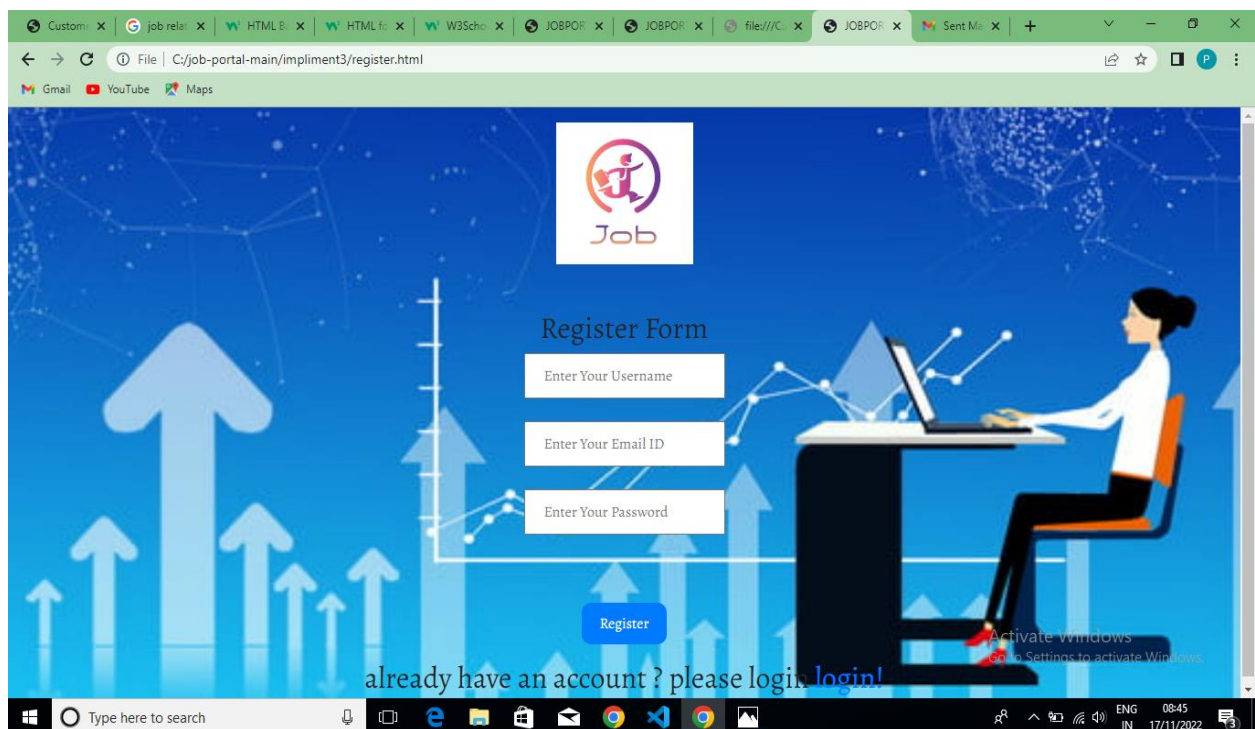
        <input type="text" name="username" placeholder="Enter Your Username"
        id="username" required></br></br>
        <input type="email" name="email" placeholder="Enter Your Email ID" id="email"
        required></br></br>
        <input type="password" name="password" placeholder="Enter Your Password"
        id="password" required></br></br>

        </br>
        </br>
        <button type="submit" id="button" class="btn btn-primary"> Register </button>
    </form>
</div>

<div class="note mt-3 text-center"> <!--Register form -->
<h2> already have an account ? please login <a href="login.html">login! </a> </h2>

</div>
</body>
</html>

```



App.py

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

app = Flask(__name__)

app.secret_key='a'
ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;PROTOCOL=TCPIP;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fxm42113;PWD=XHzCZLrUPYhyRiHW;", "", "")

@app.route('/')
def home():
    return render_template('/home.html')

@app.route('/registertemp', methods=["POST", "GET"])
def registertemp():
    return render_template("register.html")

@app.route('/uploaddata', methods = ['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        firstname = request.form['firstname']
        lastname = request.form['lastname']
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        address = request.form['address']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM userss WHERE username = ?')
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        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:
            prep_stmt = ibm_db.prepare(conn, 'INSERT INTO userss(firstname, lastname, username, email, password, address) VALUES(?, ?, ?, ?, ?, ?)')
            ibm_db.bind_param(prep_stmt, 1, firstname)
            ibm_db.bind_param(prep_stmt, 2, lastname)
            ibm_db.bind_param(prep_stmt, 3, username)
            ibm_db.bind_param(prep_stmt, 4, email)
            ibm_db.bind_param(prep_stmt, 5, password)
            ibm_db.bind_param(prep_stmt, 6, address)
            ibm_db.execute(prep_stmt)
```

```

        msg = 'Dear % s You have successfully registered!'%(username)
        return render_template('register.html',a = msg,indicator="success")
    else:
        msg = 'Please fill the form!'
        return render_template('register.html',a = msg, indicator='failure')

@app.route('/login',methods=["POST","GET"])
def login():
    return render_template("login.html")

@app.route('/logindata',methods=["POST","GET"])
def logindata():
    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 userss 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_tuple(stmt)
        if account:
            session['id'] = account[0]
            userid = account[0]
            session['username'] = account[1]
            return redirect(url_for('dashboard'))
        else:
            msg = 'Incorrect username / password !'
            return render_template('login.html', b = msg, indicator="failure")

@app.route('/home')
def dashboard():
    if 'id' in session:
        uid = session['id']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM userss WHERE id = ?')
        ibm_db.bind_param(stmt, 1, uid)
        ibm_db.execute(stmt)
        ibm_db.fetch_tuple(stmt)
        username = session['username']
        return render_template('user dashboard.html', name = username)

@app.route('/profile',methods=["POST","GET"])
def profile():
    if 'id' in session:
        uid = session['id']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM userss WHERE id = ?')
        ibm_db.bind_param(stmt, 1, uid)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_tuple(stmt)
        return
    render_template('userprofile.html',fullname=acc[1]+acc[2],username=acc[3],email=acc[4],address
=acc[6])
    return render_template('userprofile.html')

```



```

@app.route('/index',methods=["POST","GET"])
def complaint():
    if request.method == "POST":
        if 'id' in session:
            msg = ''
            uid=session['id']
            selectcategory = request.form['selectcategory']
            date = request.form['dt']
            phone = request.form['phone']
            additional = request.form['additional']
            state = request.form['state']
            email = request.form['email']
            complaint = request.form.get('complaint')
            stmt = ibm_db.prepare(conn,"SELECT * FROM userss WHERE id = ?")
            ibm_db.bind_param(stmt, 1, uid)
            ibm_db.execute(stmt)
            ibm_db.fetch_assoc(stmt)
            prep_stmt = ibm_db.prepare(conn,'INSERT INTO complaintdetails(uid, selectcategory,
dt , phone, additional, state, email, complaint, status) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)')
            ibm_db.bind_param(prepare_stmt, 1, uid)
            ibm_db.bind_param(prepare_stmt, 2, selectcategory)
            ibm_db.bind_param(prepare_stmt, 3, date)
            ibm_db.bind_param(prepare_stmt, 4, phone)
            ibm_db.bind_param(prepare_stmt, 5, additional)
            ibm_db.bind_param(prepare_stmt, 6, state)
            ibm_db.bind_param(prepare_stmt, 7, email)
            ibm_db.bind_param(prepare_stmt, 8, complaint)
            ibm_db.bind_param(prepare_stmt, 9, 'pending')
            ibm_db.execute(prepare_stmt)
            msg = 'You have successfully registered your complaint'
            return render_template('index.html',a = msg)
        return render_template('index.html')
@app.route('/view',methods=["POST","GET"])
def view():
    if 'uid' in session:
        uid = session['uid']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM complaintdetails WHERE uid = ?')
        ibm_db.bind_param(stmt, 1, uid)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_tuple(stmt)
        return render_template('comphistory.html')
    return render_template('comphistory.html')
@app.route('/comphistory',methods=['POST','GET'])
def compview():
    if 'id' in session:
        uid=session['id']
        stmt = ibm_db.prepare(conn,'SELECT * FROM complaintdetails WHERE id = ?')
        ibm_db.bind_param(stmt, 1, uid)
        ibm_db.execute(stmt)
        comp = ibm_db.fetch_assoc(stmt)
        return render_template('usercomphist.html',complaints = comp)

@app.route('/admin')
def admin():

```

```

        return render_template('admin.html')

@app.route('/adminpage')
def adminpage():
    return render_template('admin dashboard.html')

@app.route('/adminlog',methods=["POST","GET"])
def adminlog():
    msg = ''
    email = request.form['email']
    password = request.form['password']
    stmt = ibm_db.prepare(conn, 'SELECT * FROM admininfo WHERE email = ? and password = ?')
    ibm_db.bind_param(stmt,1, email)
    ibm_db.bind_param(stmt,2, password)
    ibm_db.execute(stmt)
    logged = ibm_db.fetch_assoc(stmt)
    if(logged):
        msg = 'successfully loggedin'
        return render_template("admin dashboard.html",a=msg)
    else:
        return render_template("admin.html",a="Incorrect email/password")

@app.route('/logout')
def logout():
    if 'id' in session:
        session.pop('id',None)
        session.pop('email',None)
        session.pop('password',None)
    return redirect(url_for('home'))

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

@app.route('/agent',methods=["POST","GET"])
def agent():
    return render_template('agent.html')

@app.route('/agentdata',methods=["POST","GET"])
def agentdata():
    msg = ''
    username = request.form['username']
    password = request.form['password']
    stmt = ibm_db.prepare(conn,'INSERT INTO agentinfo(username, password) VALUES (?, ?)')
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.bind_param(stmt, 2, password)
    ibm_db.execute(stmt)
    msg = 'Agent has been created successfully'
    return render_template('agent.html',a = msg)

if __name__ == '__main__':
    app.debug=True
    app.run(host='0.0.0.0',port=8080)

```