

Implementing Web Application

Team ID	PNT2022TMID46377
Project Name	Skill AndJob Recommender Application

Create UI To Interact With Application:

```
from flask import Flask, render_template
from flask_navigation import Navigation

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

import ibm_db
import bcrypt
conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=;PORT=;SECURITY=SSL;SSLServerCertificate=
DigiCertGlobalRootCA.crt;UID=;PWD=", "", "")

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

app = Flask(__name__)
app.secret_key=b'_5#y2L"F4Q8z\n\xec]/'

nav = Navigation(app)

nav.Bar('top', [
    nav.Item('Home','index'),
    nav.Item('About','about'),
    nav.Item('Signin','signin'),
    nav.Item('Signup','signup'),
])

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

@app.route('/navpage')
def navpage():
    return render_template("navpage.html")
```

```

@app.route('/About')
def about():
    return render_template("About.html")

@app.route('/Sign up')
def signup():
    @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(username,email,PASSWORD,rollNo) 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,hash)
            ibm_db.bind_param(prepare_stmt,4,rollNo)
            ibm_db.execute(prepare_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'))

    return render_template('signup.html')

if __name__ == '__main__':
    app.run()

```

Outputs:

A screenshot of a web browser displaying the 'Skill/Job Recommender Application' registration page. The page has a blue background with the title 'Skill/Job Recommender Application' in white. A white registration form is centered on the page. The form has a title 'Register Your Profile' and four input fields: 'Name' (placeholder: Enter Name), 'Phone number' (placeholder: Enter Phonenumber), 'Email Id' (placeholder: example@gmail.com), and 'Password' (placeholder: Enter Password). Below the fields is a blue 'Register' button. At the bottom of the form is a link 'Continue with LinkedIn' with a LinkedIn icon. The browser's address bar shows '127.0.0.1:5000/register'. The Windows taskbar at the bottom shows the date and time as 18/11/2022, 12:34.

login

127.0.0.1:5000/register

Skill/Job Recommender Application

Register Your Profile

Name Enter Name

Phone number Enter Phonenumber

Email Id example@gmail.com

Password Enter Password

Register

[Continue with LinkedIn](#)

30°C Mostly sunny 18/11/2022 12:34

A screenshot of a web browser displaying the 'Skill/Job Recommender Application' login page. The page has a blue background with the title 'Skill/Job Recommender Application' in white. A white login form is centered on the page. The form has a title 'Login Your Profile' and two input fields: 'Email Id' (placeholder: example@gmail.com) and 'Password' (placeholder: Enter Password). Below the fields is a blue 'Login' button. At the bottom of the form are two links: 'Continue with LinkedIn' with a LinkedIn icon, and 'New user?'. The browser's address bar shows '127.0.0.1:5000'. The Windows taskbar at the bottom shows the date and time as 18/11/2022, 12:37.

login

127.0.0.1:5000

Skill/Job Recommender Application

Login Your Profile

Email Id example@gmail.com

Password Enter Password

Login

[Continue with LinkedIn](#) [New user?](#)

30°C Mostly sunny 18/11/2022 12:37