

# **Project Report Format**

## **1. INTRODUCTION**

### **1.1 Project Overview**

This Project view provides an overview of the skill and job recommended for individuals interested in a career in any fields. It discusses the important role that any field plays in businesses and the various skills that are necessary for success in this field. It also outlines the different job opportunities available in any field and the different types of companies that employ any field professionals.

### **1.2 Purpose**

The project aims at connecting the employer and job seekers where employers are the source of the resources and the job seeker can find and apply for their targeted job. It allows registered users to be able to search for jobs and filter the results based on required skills, salary, experience level, etc. People can register as job seekers, build their profiles, and look for jobs matching their skill sets and they can apply directly to posted jobs.

## **2. LITERATURE SURVEY**

### **2.1 Existing problem**

Job recommender is a system that automatically returns a ranked list of suitable, prospective jobs for employees. It plays a significant role in connecting employees and employers. In order to choose a suitable algorithm to build the system, a comparison study of popular recommendation methods is conducted and reported in this paper.

### **2.2 References**

Anika Gupta, Dr. Deepak Garg "Applying Data Mining Techniques in Job Recommender System for Considering Candidate Job Preferences "International al Conference on Advances in Computing, Communications and Informatics (ICACCI) 2014.

Ronak V Patil, Sneha R Gadekar, Prashant P Chavan, Vikas G Aher, "Desktop based recommendation system for campus recruitment using MAHOUT", Multidisciplinary Journal of Research in Engineering and Technology, Volume 2,sue 2, Pg.480-485

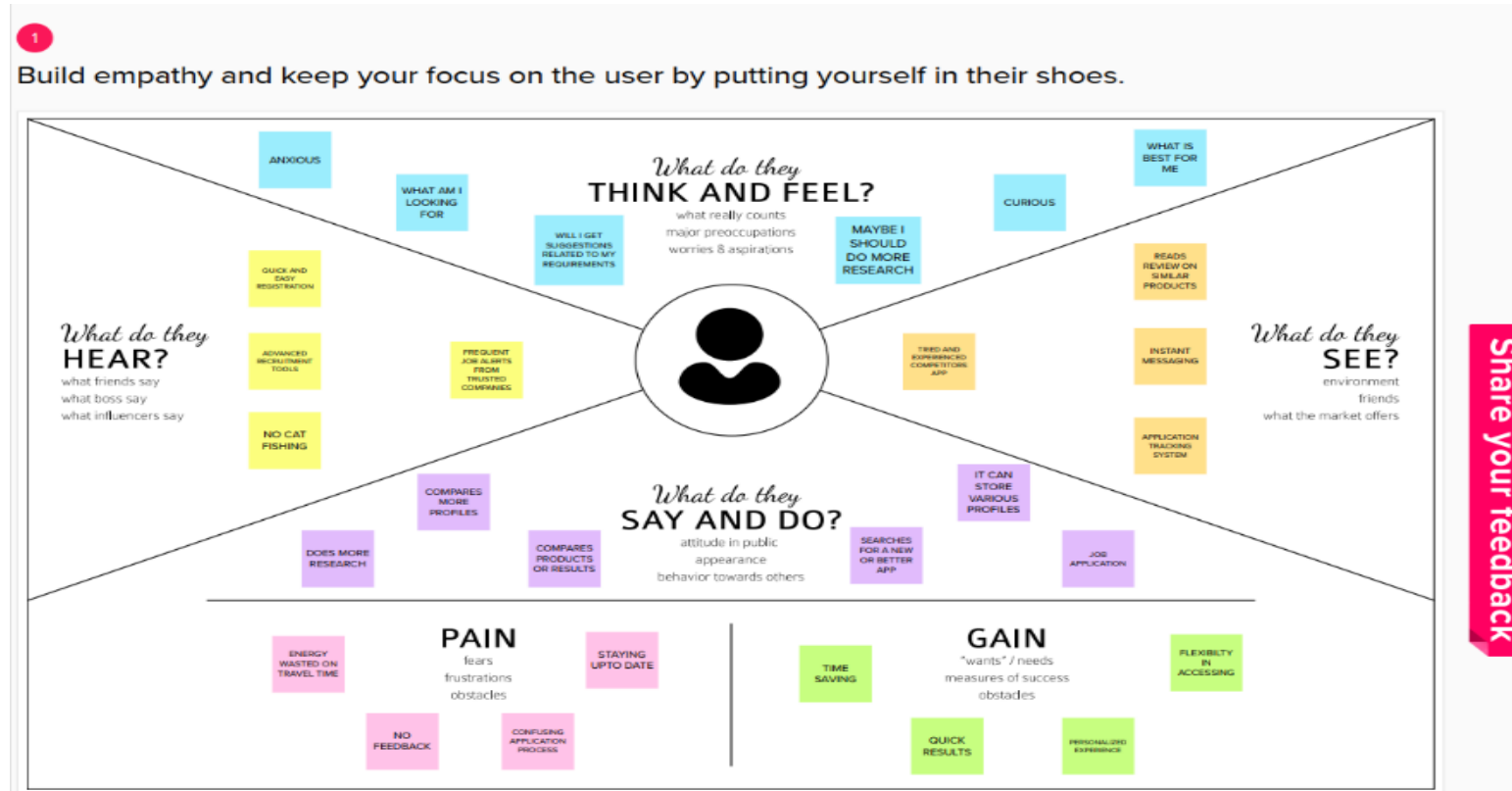
Yingya Zhang, Cheng Yang and Zhixiang NiuA, "Research of Job Recommendation System Based on Collaborative Filtering", in Seventh International Symposium on Computational Intelligence and Design, 2014 IEEE DOI 10.1109/ISCID.2014.228

### **2.3 Problem Statement Definition**

Searching of Job Vacancies based on the skill set in their profile and information about the set of preferred jobs

### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas



### 3.2 Ideation & Brainstorming

The screenshot displays the Mural web application interface for a brainstorming session. At the top, a browser window shows the URL `app.mural.co/t/ibmprojectcloudcomputing1687/m/ibmprojectcloudcomputing1687/1663217628345/2c5c9bc345cb35f809484da1ba8df5153991a595?sender=ua38f08c3c63083725...`. A pink banner below the browser indicates a "FREE PLAN" limit: "You've reached the maximum number of murals. Get unlimited murals when you upgrade your plan."

The main workspace is titled "Untitled mural" and includes a toolbar with various icons for navigation and editing. A "Facilitator" role is assigned, and a "Share" button is visible. The workspace contains a structured brainstorming template with several panels:

- Brainstorm & idea prioritization**: The first panel, which includes a grid for organizing ideas.
- Before you collaborate**: A panel with instructions for preparing for the session.
- Define your problem ...**: A panel for framing the problem as a "How Might We" statement.
- Brainstorm**: A panel for writing down ideas that address the problem statement.
- Group ideas**: A panel for clustering similar or related notes.
- Prioritize**: A panel for prioritizing ideas.


On the right side, an "Outline" panel lists the six steps of the process, each with a brief description and a "More" link. At the bottom right, a "Navigation Settings" panel is visible, showing a zoom level of 7% and a "Navigation Settings" button.


### 3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Searching of Job Vacancies based on the skill set in their profile and information about the set of preferred jobs.
2.	Idea / Solution description	Developing a website that collects data from the user and use those data to recommend job based on their preferences.
3.	Novelty / Uniqueness	Comparing recommendation list and user profile. Users are provided with a list of features out of which they can find their best.
4.	Social Impact / Customer Satisfaction	We contribute making a new dataset publicly which contains job seekers profiles and job vacancies.
5.	Business Model (Revenue Model)	The main target is to target audience ,develop new goods and services to customers and determine the place and future.
6.	Scalability of the Solution	Need to be responsive and able to react to changes as fast as possible.

### 3.4 Problem Solution fit

Problem-Solution fit canvas 2.0			Purpose / Vision		
Define CS, fit into	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span>  Unemployed candidates of age above 22, Experienced professionals who want to switch their jobs.	<b>6. CUSTOMER LIMITATIONS</b> <span>CC</span>  Network connections, possibility of scam, Limited services problem due to subscription.	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span>  SOLUTIONS: Segregation of job field PROS: Stay connected and get employed CONS: Spam messages	Explore AS,	
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span>  Creating a job recommending platform  Filtering the jobs based on skillset and experience	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span>  Fake profile which seeks money and scam people, Personal information collected by recommender raises the risk of unwanted exposure of information.	<b>7. BEHAVIOUR</b> <span>BE</span>  Creating a profile and login credentials, Searching job vacancy related to their interests and skill sets. Tracking of the application status.		
Focus on J&P, tap into BE, understand	<b>3. TRIGGERS</b> <span>TR</span>  Time saving, Job Alerts, Advertisements	<b>10. YOUR SOLUTION</b> <span>SL</span>  To develop an application that helps users to overcome their difficulty to search for jobs.	<b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span>  8.1 ONLINE Applying job, Live tracking of application status.  8.2 OFFLINE Final round of Interview process, Details Validation.	Extract online & offline CH of BE	
	<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span>  BEFORE: Stressed, Confusing application process. AFTER: Quick result, up to date information.				


Problem-Solution fit canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license Created by Daria Neprikhina / Amaltama.com



## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

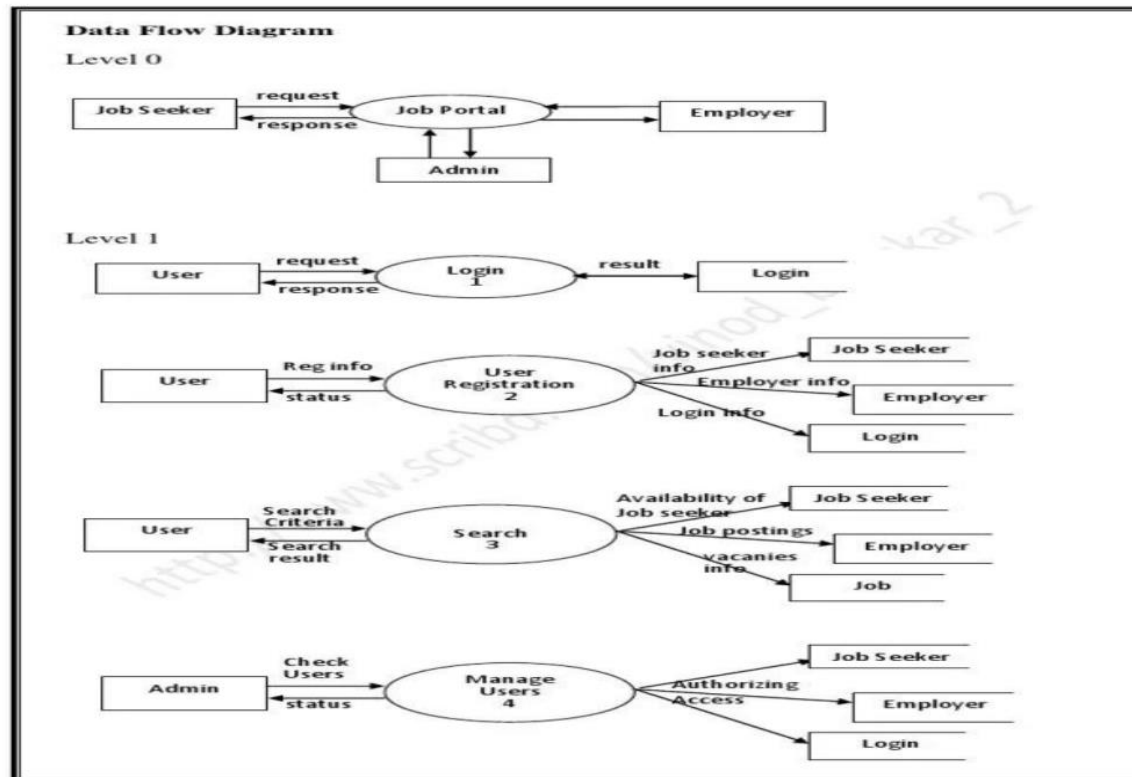
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail Registration through google form
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	User can login using login credentials – registered mail ID and password
FR-4	User profile	User should complete their profile and need to update it periodically
FR-5	User search	Job searching based on their area of interest skillset and experience
FR-6	User application	User can apply for their interested job profile

### 4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	User can login to the application and can search the job based on the skillset and apply to the ob. They can leave feedback to interview process and their experience
NFR-2	<b>Security</b>	Allow verified users to login Available security features are Mail verification, OTP verification
NFR-3	<b>Reliability</b>	Applicants can access the website effectively for job searching, less downtime of server, open source
NFR-4	<b>Performance</b>	Quick response and minimum consumption of time for any process
NFR-5	<b>Availability</b>	It can be accessed by the user at anytime and anywhere independent of the users location
NFR-6	<b>Scalability</b>	It is responsive and has the ability to react to changes as soon as possible

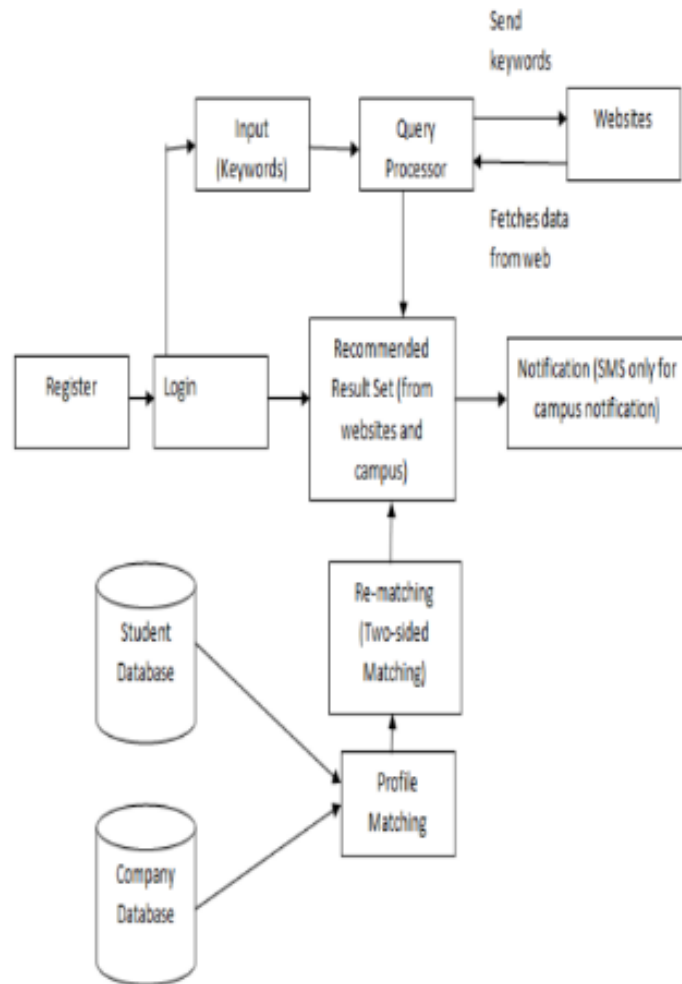
## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams



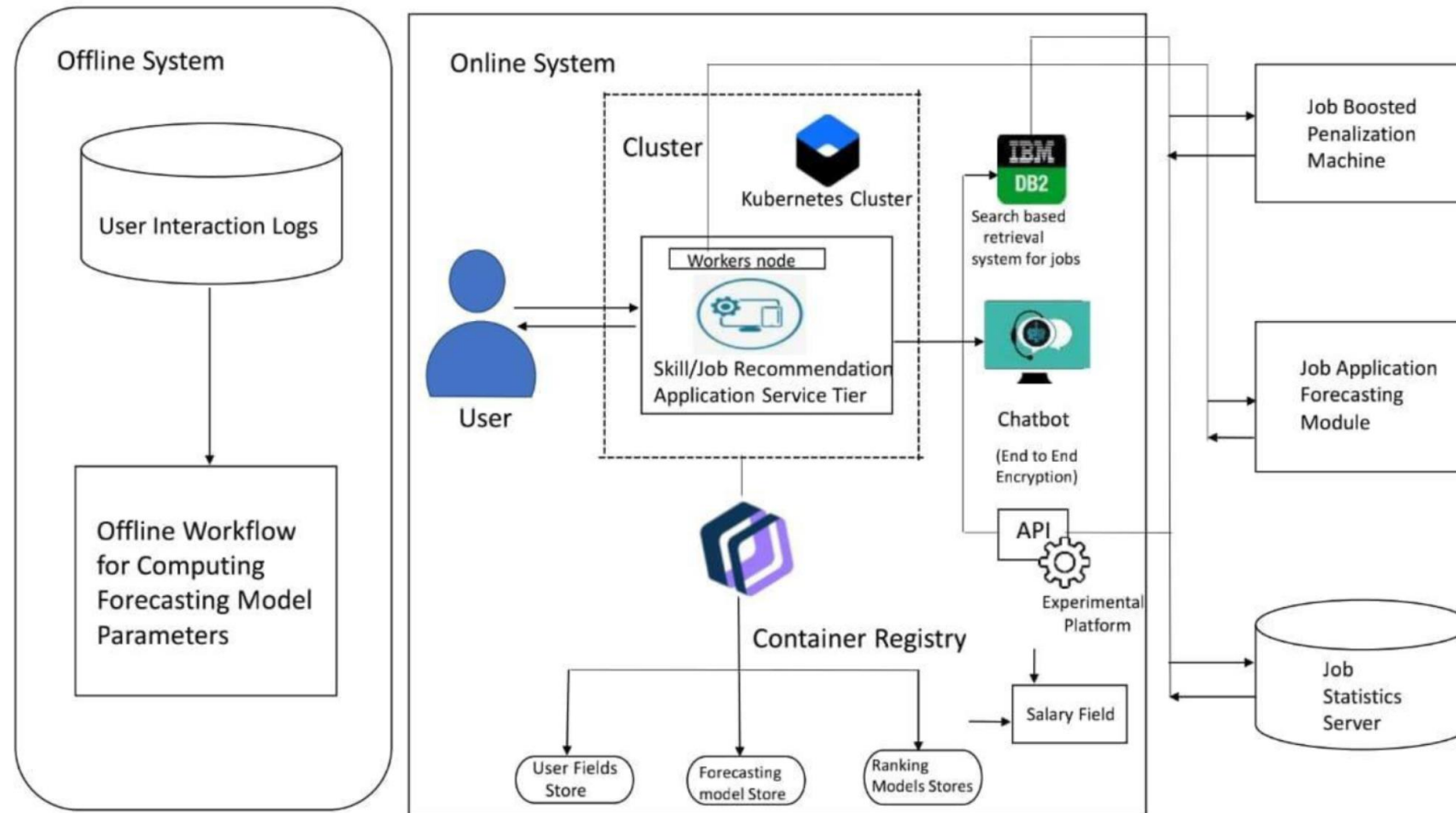
## 5.2 Solution & Technical Architecture

### Solution Architecture





## Technical Architecture



### 5.3 User Stories

#### User Stories :

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation email & click confirm	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	Create a model set that contains those models, then assign it to a role.	Assign that group to the appropriate roles on the Roles page	High	Sprint-1
Customer (Web user)	Identity-Aware	USN-7	Open, public access, User-authenticated access, Employee-restricted access.	Company public website. App running on the company intranet. App with access to customer private information.	High	Sprint-1
Customer Care Executive	Communication	USN-8	A customer care executive is a professional responsible for communicating the how's and why's regarding service expectations within a company.	For how to tackle customer queries.	Medium	Sprint-1
Administrator	Device management	USN-9	You can Delete/Disable/Enable devices in Azure Active Directory but you cannot Add/Remove Users in the directory.	Ease of use.	Medium	Sprint-1

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

<b>Sprint</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Story Points</b>	<b>Priority</b>	<b>Team Members</b>
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming mypassword.	1	High	Varuna Priya. S
Sprint-1			As a user, I will receive confirmation email once I have registered for the application	1	Medium	Varuna Priya. S
Sprint-2	Login	USN-2	As a user, I can login the application by entering email & password	1	High	Shiny Shalynn I M
Sprint-2			As a user, I can search for job vacancies.	1	Medium	Shiny Shalynn I M
Sprint-3	Dashboard	USN-3	As a user, I can apply for jobs based on criteria	1	High	Sudarshni Ramesh

<b>Sprint</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Story Points</b>	<b>Priority</b>	<b>Team Members</b>
Sprint-3			The applicants will be shortlisted based on job description.	1	Medium	Sudarshni Ramesh
Sprint-4	Dashboard	USN-4	The candidates will be notified for the interview process.	1	High	Sathiyapriya T
Sprint-4			The selected candidates will be notified to their respective e-mail.	1	Medium	Sathiyapriya T

## 6.2 Sprint Delivery planning:

Project Tracker, Velocity & Burndown Chart:

<b>Sprint</b>	<b>Total Story Points</b>	<b>Duration</b>	<b>Sprint Start Date</b>	<b>Sprint End Date (Planned)</b>	<b>Story Points Completed (as on Planned End Date)</b>	<b>Sprint Release Date (Actual)</b>
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

## 6.2 Reports from JIRA

$$AV = \frac{\textit{sprint duration}}{\textit{velocity}} = \frac{20}{10} = 2$$

## 7 CODING & SOLUTIONING (Explain the features added in the project along with code)

### 7.1 Feature 1

<script>

```
window.watsonAssistantChatOptions = {  
  integrationID: "f91ff97d-0b71-4b82-a355-cac0e408986e", // The ID of this integration.  
  region: "au-syd", // The region your integration is hosted in.  
  serviceInstanceID: "9509f524-94c8-4688-970f-7fdb98f15b39", // The ID of your service instance.  
  onLoad: function(instance) { instance.render(); }  
};  
  
setTimeout(function(){  
  const t=document.createElement('script');  
  t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') +  
  "/WatsonAssistantChatEntry.js";  
  document.head.appendChild(t);  
});  
</script>
```

### 7.2 Feature 2

```
import ibm_db
```

```

try:
    conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=2d46b6b4-cbf6-40eb-bbce-
        6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=32328;SECURITY=SSL;SSlServerCertificate=DigiCertGlobalRootCA.
        crt;UID=tzp28629;PWD=TVm24LZqNge08eTF;", ", ")
    print("db is connected")
except:
    print("db is not connected")

```

### 7.3 Database Schema

```

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders

body = '''
JOB RECOMMENDER

Hello first_name,

Thank you for creating a account in this App we hope to serve you the best.
In the last years, job recommender systems have become popular since they
successfully reduce information overload by generating personal-ized job
suggestions. Although in the literature exists a variety of techniques and
strategies used as part of job recommender systems, most of them fail to
recommending job vacancies that fit properly to the job seekers profiles.
Thus, the contributions of this work are threefold, we: i) made publicly
available a new dataset formed by a set of job seekers profiles and a set
of job vacancies collected from different job search engine sites; ii) put
forward the proposal of a framework for job recommendation based on
professional skills of job seekers; and iii) carried out an evaluation to

```

quantify empirically the recommendation abilities of two state-of-the-art methods, considering different configurations, within the proposed framework. We thus present a general panorama of job recommendation task aiming to facilitate research and real-world application design regarding this important issue.

Sincerely,

employee\_name

'''

sender = 'shruthiramesh2001@gmail.com'

password = 'kabchcujcuxrvdve'

receiver = 'shruthiramesh2001@gmail.com'

message = MIMEMultipart()

message['From'] = sender

message['To'] = receiver

message['Subject'] = 'This email has an attacment, a pdf file'

message.attach(MIMEText(body, 'plain'))

qimage = "img4.jpg"

binary\_pdf = open(qimage, 'rb')

payload = MIMEBase('application', 'octate-stream', Name=qimage)

payload.set\_payload((binary\_pdf).read())

encoders.encode\_base64(payload)

payload.add\_header('Content-Decomposition', 'attachment', filename=qimage)

message.attach(payload)

```
session = smtplib.SMTP('smtp.gmail.com', 587)

session.starttls()

session.login(sender, password)

text = message.as_string()
session.sendmail(sender, receiver, text)
session.quit()
print('Mail Sent')
```

## 8 TESTING

### 8.1 User Acceptance Testing

#### Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Skills/Job Recommender.Application project at the time of the release to User Acceptance Testing (UAT).

#### Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

## 9 RESULTS

The project has been completed as we expected. We ensured that Database was designed and well connected to our project.The Expected results were gotten.



## 10 ADVANTAGES & DISADVANTAGES

Advantages:

- 1) Easy recommendations make less search and sometimes end up un good deals
- 2) User reviews will give accurate information, this is also an advantage if you purchase online as you can see other reviews too, most of the times honest
- 3) Speed up the process of decision and purchase based on the previous statistics

Disadvantages:

- 1) If the system recommends products with bias, then customer will be landing into wrong deals
- 2) Chances are that some websites may suggest products wrongly based on analysis of little information gathered

## 11 CONCLUSION

The application has been developed to make job search easier .The application that we have developed is user friendly .User can find a job based on their skillset in the short period of time. The jobseeker certainly get benefit by using this application. In the addition, Chatbot Has been implemented with the help of IBM watson . The chatbot helps jobseeker and organization when they experience the difficulties.

## 12 FUTURE SCOPE

Future directions of our work will focus on performing a more exhaustive evaluation considering a greater amount of methods and data as well as a comprehensive evaluation of the impact of each professional skill of a job seeker on the received job recommendation.

## 13 APPENDIX

Source Code

```
import ibm_db
```

```
try:
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=32328;SECURITY=SSL;SSlServerCertificate=DigiCertGlobalRootCA.
```

```
    crt;UID=tzp28629;PWD=TVm24LZqNge08eTF;"," ")
print("db is connected")
except:
    print("db is not connected")
```

```
index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <script
        src="https://kit.fontawesome.com/64d58efce2.js"
        crossorigin="anonymous"
    ></script>
    <link rel="stylesheet" href="{ { url_for('static', filename='Styles/style.css') } }" />
    <title>Sign in & Sign up Form</title>
</head>
<body>
    <script src="{ { url_for('static', filename='Js/hide.js') } }"></script>

    <div class="container">
        <div class="forms-container">
```

```
<div class="signin-signup">
```

```
<form action="/login" class="sign-in-form" method="post">
```

```
<h2 class="title">LOG IN
```

```
</h2>
```

```
<div class="cont">
```

```
<h6 id="box">
```

```
{ {msg} }
```

```
</h6>
```

```
</div>
```

```
<div class="input-field">
```

```
<i class="fas fa-envelope"></i>
```

```
<input type="email" placeholder="Email" / name="email" required>
```

```
</div>
```

```
<div class="input-field">
```

```
<i class="fas fa-lock"></i>
```

```
<input type="password" placeholder="Password" / name="password"required>
```

```
</div>
```

```
<input type="submit" value="Login" class="btn solid"/>
```

</form>

```
<form action="/register" class="sign-up-form" method="post" name="sign" onsubmit="return validateform()">
  <h2 class="title">REGISTER</h2>
  <div class="input-field">
    <i class="fas fa-user"></i>
    <input type="text" placeholder="Username" / name="username" required>
  </div>
  <div class="input-field">
    <i class="fas fa-envelope"></i>
    <input type="email" placeholder="Email" / name="email" required>
  </div>
  <div class="input-field">
    <i class="fa fa-phone"></i>
    <input type="tel" placeholder="Phone" / name="phone" required>
  </div>
  <div class="input-field">
    <i class="fas fa-lock"></i>
    <input type="password" placeholder="Password" / name="password" required>
```

```

    </div>
    <input type="submit" class="btn" value="Sign up" />
  </form>
</div>
</div>

<div class="panels-container">
  <div class="panel left-panel">
    <div class="content">
      <h3 id="changeText">For Register?
      <script src="{ { url_for('static', filename='Js/head.js') } }"></script></h3><br>

      <button class="btn transparent" id="sign-up-btn">
        Sign up
      </button>
    </div>
    
  </div>
  <div class="panel right-panel">
    <div class="content">
      <h3 id="changeText1">
      <script src="{ { url_for('static', filename='Js/reg.js') } }"></script>
      </h3> <br>
      <p style="font-weight:600;">

```

Already have an account?

</p>

<button class="btn transparent" id="sign-in-btn">

Sign in

</button>

</div>



</div>

</div>

</div>

<script src="{ { url\_for('static', filename='Js/app.js') } }"></script>

<script src="{ { url\_for('static', filename='Js/sign.js') } }"></script>

</body>

</html>

App.py

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

```
import db
```

```
import os
```

```
from db import ibm_db
```

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

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

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

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

@app.route('/login', methods = ['GET', 'POST'])
def login():
    global userid
    msg = ""

    if request.method == 'POST':
        email = request.form['email']
```

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

```
sql = "SELECT * FROM USERDETAILS WHERE email = ? AND password = ?"
```

```
stmt = ibm_db.prepare(db.conn,sql)
```

```
ibm_db.bind_param(stmt,1,email)
```

```
ibm_db.bind_param(stmt,2,password)
```

```
ibm_db.execute(stmt)
```

```
account = ibm_db.fetch_assoc(stmt)
```

```
print(account)
```

```
if account:
```

```
    return render_template("home.html",msg=msg)
```

```
else:
```

```
    msg = 'Invalid details. Please check the Email ID - Password combination.!' 
```

```
    return render_template("index.html",msg=msg)
```



```

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        phone = request.form['phone']
        password = request.form['password']
        sql = "SELECT * FROM USERDETAILS WHERE email = ?"
        stmt = ibm_db.prepare(db.conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg='Job Recommender Account Already exist.kindly login!'
            return render_template("index.html",msg=msg)

        else:
            sql="INSERT INTO USERDETAILS(USERNAME,EMAIL,PHONE,PASSWORD) VALUES('{0}','{1}','{2}','{3}')"
            res = ibm_db.exec_immediate(db.conn,sql.format(username,email,phone,password))
            msg = "Your Job Recommender account successfully registered!"
            return render_template("index.html",msg=msg)

```

```
if(__name__=='__main__'):
```

```
    app.run(host='0.0.0.0',debug=True)
```

```
style.css
```

```
@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700;800&display=swap");
```

```
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}
```

```
body,  
input {  
    font-family: "Poppins", sans-serif;  
}
```

```
.container {  
  position: relative;  
  width: 100%;  
  background-color:aliceblue;  
  min-height: 100vh;  
  overflow: hidden;  
}
```

```
.forms-container {  
  position: absolute;  
  width: 100%;  
  height: 100%;  
  top: 0;  
  left: 0;  
}
```

```
.signin-signup {  
  position: absolute;  
  top: 50%;  
  transform: translate(-50%, -50%);  
  left: 75%;  
  width: 50%;  
  transition: 1s 0.7s ease-in-out;  
  display: grid;
```

```
    grid-template-columns: 1fr;  
    z-index: 5;  
}
```

```
form {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    flex-direction: column;  
    padding: 0rem 5rem;  
    transition: all 0.2s 0.7s;  
    overflow: hidden;  
    grid-column: 1 / 2;  
    grid-row: 1 / 2;  
}
```

```
form.sign-up-form {  
    opacity: 0;  
    z-index: 1;  
}
```

```
form.sign-in-form {  
    z-index: 2;  
}
```

```
.title {  
  font-size: 2.2rem;  
  color: #444;  
  margin-bottom: 10px;  
}
```

```
.title1 {  
  font-size: 1.6rem;  
  color: #444;  
  margin-bottom: 10px;  
}
```

```
.input-field {  
  max-width: 380px;  
  width: 100%;  
  background-color: #f0f0f0;  
  margin: 10px 0;  
  height: 55px;  
  border-radius: 55px;  
  display: grid;  
  grid-template-columns: 15% 85%;  
  padding: 0 0.4rem;  
  position: relative;
```

```
}
```

```
.input-field i {  
  text-align: center;  
  line-height: 55px;  
  color: black;  
  transition: 0.5s;  
  font-size: 1.1rem;  
}
```

```
.input-field input {  
  background: none;  
  outline: none;  
  border: none;  
  line-height: 1;  
  font-weight: 600;  
  font-size: 1.1rem;  
  color: #333;  
}
```

```
.input-field input::placeholder {  
  color: #aaa;  
  font-weight: 500;  
}
```

```
.btn {  
  width: 150px;  
  background-color: #5995fd;  
  border: none;  
  outline: none;  
  height: 49px;  
  border-radius: 49px;  
  color: #fff;  
  text-transform: uppercase;  
  font-weight: 600;  
  margin: 10px 0;  
  cursor: pointer;  
  transition: 0.5s;  
}
```

```
.btn:hover {  
  background-color: #4d84e2;  
}
```

```
.panels-container {  
  position: absolute;  
  height: 100%;
```

```
width: 100%;  
top: 0;  
left: 0;  
display: grid;  
grid-template-columns: repeat(2, 1fr);  
}
```

```
.container:before {  
  content: "";  
  position: absolute;  
  height: 2000px;  
  width: 2000px;  
  top: -10%;  
  right: 48%;  
  transform: translateY(-50%);  
  background-image: linear-gradient(-45deg, #4481eb 0%, #04befe 100%);  
  transition: 1.8s ease-in-out;  
  border-radius: 50%;  
  z-index: 6;  
}
```

```
.image {  
  width: 100%;  
  transition: transform 1.1s ease-in-out;
```



```
    transition-delay: 0.4s;  
  }
```

```
.panel {  
  display: flex;  
  flex-direction: column;  
  align-items: flex-end;  
  justify-content: space-around;  
  text-align: center;  
  z-index: 6;  
}
```

```
.left-panel {  
  pointer-events: all;  
  padding: 3rem 17% 2rem 12%;  
}
```

```
.right-panel {  
  pointer-events: none;  
  padding: 3rem 12% 2rem 17%;  
}
```

```
.panel .content {  
  color: #fff;
```

```
transition: transform 0.9s ease-in-out;  
transition-delay: 0.6s;  
}
```

```
.panel h3 {  
  font-weight: 600;  
  line-height: 1;  
  font-size: 1.5rem;  
}
```

```
.panel p {  
  font-size: 0.95rem;  
  padding: 0.7rem 0;  
}
```

```
.btn.transparent {  
  margin: 0;  
  background: none;  
  border: 2px solid #fff;  
  width: 130px;  
  height: 41px;  
  font-weight: 600;  
  font-size: 0.8rem;  
}
```

```
.right-panel .image,  
.right-panel .content {  
  transform: translateX(800px);  
}
```

```
/* ANIMATION */
```

```
.container.sign-up-mode:before {  
  transform: translate(100%, -50%);  
  right: 52%;  
}
```

```
.container.sign-up-mode .left-panel .image,  
.container.sign-up-mode .left-panel .content {  
  transform: translateX(-800px);  
}
```

```
.container.sign-up-mode .signin-signup {  
  left: 25%;  
}
```

```
.container.sign-up-mode form.sign-up-form {  
  opacity: 1;
```

```
    z-index: 2;
}
```

```
.container.sign-up-mode form.sign-in-form {
    opacity: 0;
    z-index: 1;
}
```

```
.container.sign-up-mode .right-panel .image,
.container.sign-up-mode .right-panel .content {
    transform: translateX(0%);
}
```

```
.container.sign-up-mode .left-panel {
    pointer-events: none;
}
```

```
.container.sign-up-mode .right-panel {
    pointer-events: all;
}
```

```
@media (max-width: 870px) {
    .container {
        min-height: 800px;
    }
}
```

```
    height: 100vh;
  }
  .signin-signup {
    width: 100%;
    top: 95%;
    transform: translate(-50%, -100%);
    transition: 1s 0.8s ease-in-out;
  }
```

```
.signin-signup,
.container.sign-up-mode .signin-signup {
  left: 50%;
}
```

```
.panels-container {
  grid-template-columns: 1fr;
  grid-template-rows: 1fr 2fr 1fr;
}
```

```
.panel {
  flex-direction: row;
  justify-content: space-around;
  align-items: center;
  padding: 2.5rem 8%;
```

```
    grid-column: 1 / 2;  
}
```

```
.right-panel {  
    grid-row: 3 / 4;  
}
```

```
.left-panel {  
    grid-row: 1 / 2;  
}
```

```
.image {  
    width: 200px;  
    transition: transform 0.9s ease-in-out;  
    transition-delay: 0.6s;  
}
```

```
.panel .content {  
    padding-right: 15%;  
    transition: transform 0.9s ease-in-out;  
    transition-delay: 0.8s;  
}
```

```
.panel h3 {
```

```
font-size: 1.2rem;  
}
```

```
.panel p {  
font-size: 0.7rem;  
padding: 0.5rem 0;  
}
```

```
.btn.transparent {  
width: 110px;  
height: 35px;  
font-size: 0.7rem;  
background:red;  
}
```

```
.container:before {  
width: 1500px;  
height: 1500px;  
transform: translateX(-50%);  
left: 30%;  
bottom: 68%;  
right: initial;  
top: initial;  
transition: 2s ease-in-out;
```

```
}
```

```
.container.sign-up-mode:before {  
  transform: translate(-50%, 100%);  
  bottom: 32%;  
  right: initial;  
}
```

```
.container.sign-up-mode .left-panel .image,  
.container.sign-up-mode .left-panel .content {  
  transform: translateY(-300px);  
}
```

```
.container.sign-up-mode .right-panel .image,  
.container.sign-up-mode .right-panel .content {  
  transform: translateY(0px);  
}
```

```
.right-panel .image,  
.right-panel .content {  
  transform: translateY(300px);  
}
```

```
.container.sign-up-mode .signin-signup {
```



```
    top: 5%;  
    transform: translate(-50%, 0);  
  }  
}
```

```
@media (max-width: 570px) {  
  form {  
    padding: 0 1.5rem;  
  }  
}
```

```
.image {  
  display: none;  
}  
.panel .content {  
  padding: 0.5rem 1rem;  
}  
.container {  
  padding: 1.5rem;  
}
```

```
.container:before {  
  bottom: 72%;  
  left: 50%;  
}
```

```
.container.sign-up-mode:before {  
    bottom: 28%;  
    left: 50%;  
}  
}
```

```
#changeText1,#changeText{  
    width: 600px;  
}
```

```
#box{  
    color: red;  
}  
.cont{  
    width: 500px;  
    height: 10px;  
    padding-left: 100px;  
}
```

Home.html

```
<!DOCTYPE html>  
<html>  
    <head>
```

```

<link rel="stylesheet" href="{ {url_for('static' , filename='Styles/indexstyle.css')}} ">
<script>
    window.watsonAssistantChatOptions = {
        integrationID: "f91ff97d-0b71-4b82-a355-cac0e408986e", // The ID of this integration.
        region: "au-syd", // The region your integration is hosted in.
        serviceInstanceID: "9509f524-94c8-4688-970f-7fdb98f15b39", // The ID of your service instance.
        onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
        const t=document.createElement('script');
        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') +
        "/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });
</script>
</head>
<body>

    <!--navbar-->
    <nav class="navbar">
        <h1>Place where you can get your dream job</h1>
        <div class="navbar-menu" >
            <a href="#job">Jobs</a>
            <a href="#companies">Companies</a>

```

```
<a href="#blog">Blog</a>
<a href="#login">Login</a>
</div>
<div class="menu-toggle">

    <span class="bar"></span>
    <span class="bar"></span>
    <span class="bar"></span>
    <span class="bar"></span>
</div>
</nav>
<header>
    <h1 class="header-title">JOB RECOMMENDER</h1>
</header>

</body>
</html>
```

Job.html

```
<head>
    <link rel="stylesheet" href="../static/style.css">
    <link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Rancho&effect=shadow-multiple">
```

```
<title>Python Website Demo</title>
</head>
<body>
  <br><br><br><br>
  <center>
    <h2>Browse all the Jobs</h2><br>
    <div id="search">
      <frame>
        <form method="POST" name=form action="/browse/searchjob">
          <select type="text" name="searchopt">
            <option value="JOBTITLE">JOBTITLE</option>
            <option value="JOBTYPE">JOBTYPE</option>
            <option value="EXPERIENCE">EXPERIENCE</option>
            <option value="KEYSKILL">KEYSKILL</option>
            <option value="LOCATION">LOCATION</option>
            <option value="SALARY">SALARY</option>
            <option>OTHER</option>
          </select>
          <input type="text" name="srctitle">
          <button class="button" type="submit" value="submit" id="button">Search Now</button>
        </form>
      </div>
      <table>
        <tr style="height: 60px; background-color: #04aec4; color: white;">
```

<td> JOBTITLE </td> <td> JOBTYP </td> <td> EXPERIENCE </td> <td> KEYSKILL </td> <td> LOCATION </td> <td> SALARY </td> <td> APPLY </td> </tr>
---

```
<tr>
  <td>senior developer</td>
  <td>frontend</td>
  <td>3 -5 years</td>
  <td>hands on skill in front end technology,good in communication,react,</td>
  <td>chennai,mumbai</td>
  <td>6 LPA</td>
  <td>
    <button type="submit" >APPLY</button>
  </td>
</tr>
```

```
</table>
</frame>
</center>
```

```
<style type="text/css">
body{
  background-color: white;
  color: black;
}
table{
  border-collapse: separate;
  box-shadow:2px 3px 3px rgb(165, 168, 166);
```

```
}  
h2{  
    margin-right: 190px;  
}  
  
table tr{  
    height: 90px;  
    background-color: white;  
}  
table tr td{  
    width: 200px;  
    text-align: center;  
}  
#search{  
    margin-left: -130px;  
}
```

</style>

Indexstyle.css

```
:root{  
    --primaryColor:rgba(27, 206, 206, 0.911);  
    --whileColor:rgba(225, 231, 231, 0.911);
```



```
--blackColor:rgb(28, 29, 48);
--greyColor:#f6f8fa;
--labelColor:#092a47;
--yellowColor:rgba(240, 240, 9, 0.856);

}
*{
  margin:0;
  padding:0;
  box-sizing:0;
  outline:none;
  font-family:'Nunito', sans-serif;

}
body{
  width:100%;
  height:auto;
}
h1{
  line-height: 1.5;
  color:aqua;
}
.navbar{
  width:100%;
```

```
height:60px;
display:flex;
justify-content:space-between;
top:0;
left:0;
padding:1rem 2.5rem;
z-index:10;
color:var(--whiteColor);
background:black;
}
.navbar-menu{

width:400px;
display:flex;
justify-content: space-between;

}
.navbar-menu >a {
text-decoration:none;
}
.menu-toggle{
display:block;
}
.menu-toggle .bar{
```

```
width:15px;
height:2px;
margin:3px auto ;
transition:all 0.3 ease-in-out;
background-color:var(--whitecolor);
cursor:pointer;
}
header{
    width:100%;
    height:80%;
    background-image: url("images/img.jpg");
    background-position:center;
    background-repeat: no-repeat;
    position:relative;
}
```

Job.css

```
*{
    margin: 0;
    padding: 0;
}
```

```
.wrapper{  
  width: 100%;  
  height: 80px;
```

```
}
```

```
header{  
  height: 125vh;  
  -webkit-background-size: cover;  
  background-size: cover;  
  background-position: center center;
```

```
}
```

```
.nav-area{  
  float: right;  
  list-style: none;  
  margin-top: 25px;  
  margin-right: 150px;  
}
```

```
.nav-area li{
```

```
display: inline-flex;  
cursor: pointer;  
}
```

```
.nav-area li a{  
  color: rgb(0, 0, 0);  
  text-decoration: none;  
  padding: 5px 25px;  
  border-radius: 15px;  
  font-family:'Arial';  
  font-size: large;  
  font-weight: bolder;  
  background: rgb(255, 194, 39);  
  
}
```

```
.nav-area li a:hover{  
  background: rgb(0, 0, 0);  
  color: rgb(255, 255, 255) ;  
  
}
```

```
.logo img{
```

```
margin-top: 10px;
width: 100px;
margin-left: 300px;
float: left;
}
```

```
.logo-end img{
margin-top: 15px;
width: 70px;
margin-right: 15px;
margin-bottom: 10px;
margin-left: 600px;
float: left;
}
```

```
.card {

box-shadow: 0 4px 8px 0 rgba(190, 95, 23, 0.2);
transition: 0.3s;
width: 80%;
background: -webkit-linear-gradient(bottom,#ffffff, #ffffff );

}
```

```
.card1 {
```

```
  box-shadow: 0 4px 8px 0 rgba(190, 95, 23, 0.2);
```

```
  width: 70%;
```

```
  background: -webkit-linear-gradient(bottom,#ffffff, #ffffff );
```

```
  height: 50%;
```

```
}
```

```
table{
```

```
  margin: 30px;
```

```
}
```

```
.card:hover {
```

```
  box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2);
```

```
}
```

```
section .contentBx{
```

```
  position: absolute;
```

```
  justify-content: center;
```

```
  align-items: center;
```

```
  margin: 5% 50%;
```

```
  width: 50%;
```

```
height: 100%;
```

```
}
```

```
section .contentBx .formBx{
```

```
width: 70%;
```

```
padding: 30px 10px;
```

```
margin:auto;
```

```
}
```

```
section .contentBx .formBx h2{
```

```
color:rgb(0, 0, 0);
```

```
font-weight: 600;
```

```
font-size: 1.6em;
```

```
text-align: center;
```

```
font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
```

```
text-transform: uppercase;
```

```
margin-bottom: 20px;
```

```
margin-left: 100px;
```

```
border-bottom: 4px solid rgb(251, 186, 20);
```

```
display: inline-block;
```



```
    letter-spacing: 1px;  
}
```

```
section .contentBx .formBx .inputBx{  
    margin-bottom: 20px;  
  
}
```

```
section .contentBx .formBx .inputBx span{  
    margin-bottom: 10px;  
    font-size: 14px;  
    display: inline-block;  
    margin-left: 5px;  
    color:rgb(0, 0, 0);  
    font-weight: bolder;  
    font-family:'Arial';  
    letter-spacing: 1px;  
  
}
```

```
section .contentBx .formBx .inputBx input{  
    width: 100%;  
    padding: 10px 10px;
```

```
outline: none;
font-weight: 400;
border: 2px solid rgb(255, 191, 0);
font-family:'Arial';
font-size: 14px;
letter-spacing: 1px;
color:rgb(0, 0, 0);
background: white;
border-radius: 30px;

}

section .contentBx .formBx .inputBx input[type="submit"]{

background: rgb(251, 186, 20);
color:black;
outline: none;
border: none;
margin-left: 27.5px;
font-size: 18px;
font-weight: bolder;
font-family:'Arial';
cursor: pointer;
```

```
}
```

```
section .contentBx .formBx .inputBx input[type="submit"]:hover{  
    background: rgb(255, 210, 96);  
}
```

```
section .contentBx .formBx .remember{  
    margin-bottom: 20px;  
    color:rgb(0, 0, 0);  
    font-weight: 400;  
    font-size: 14px;  
    font-weight: bolder;  
    margin-left: 8px;  
    cursor: pointer;  
    font-family:'Arial';  
}
```

```
section .contentBx .formBx p{  
    color:rgb(0, 0, 0);  
    font-family:'Arial';  
    font-size: 14px;  
    margin-left: 100px;  
    font-weight: bolder;
```

```
}
```

```
section .contentBx p a{  
    margin-left: 2px;  
    color: rgb(0, 98, 255);  
    font-weight: bolder;  
    font-family:'Arial';  
    margin-left: 3px;  
    font-size: 14px;  
    text-decoration: none;
```

```
}
```

```
footer {  
    position:relative;  
    max-height: 50%;  
    width: 100%;  
    height: 300px;  
    left: 0;  
    bottom: 0;  
    background-color: rgb(250, 203, 14);  
}
```

```
.intro h5{  
    text-align: center;  
    color: rgb(0, 0, 0);  
    font-size: 17px;  
    text-decoration: none;  
    line-height: 30px;  
  
}
```

```
.intro h5 a{  
    text-decoration: none;  
    color: rgb(33, 4, 252);  
    font-weight: bold;  
  
}
```

```
#back{  
    background-color: rgb(148, 137, 137);  
  
}
```

```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  margin: 20px 200px;  
  gap: 20px;  
  background-color: rgb(6, 121, 244);  
  padding: 25px;  
  position: relative;  
  height: 260px;  
  text-align: center;  
}
```

```
.grid-container .grid {  
  background-color: white;  
  text-align: center;  
  padding: 50px 0;  
  font-size: 20px;  
  color: rgb(0, 0, 0);  
}
```

```
#start{  
  height: 2.5px;  
  background-color: rgb(6, 121, 244);  
}
```

```
.job-sign{  
  width: 100%;  
  left: 0;  
  height: 260px;  
  bottom: 0;  
  margin-top: 5px;  
  margin-bottom: 5px;  
  background-color: rgb(6, 121, 244);  
}
```

```
.job-sign h1 {  
  font-size: 37.5px;  
  align-items: center;  
  margin-left: 500px;  
  color: rgb(255, 196, 0);  
}
```

```
#sign-center{  
  border: 0px solid rgb(6, 121, 244);  
  padding: 12.5px 0px;  
  text-decoration: none;  
  text-transform: uppercase;  
  font-size: 19px;  
  width: 8%;
```

```
margin-top: 20px;
margin-left: 745px;
font-weight: bolder;
cursor: pointer;
color: rgb(0, 0, 0);
background:rgb(251, 202, 4);
border-radius: 20px;
}
```

```
.s-media img{
margin-right: 5px;
margin-left: 10px;
float: left;
}
```

```
.contri{
float: left;
margin-left: 450px;
margin-top: 30px;
}
```

```
.contri h4, a{

margin-bottom: 2px;
margin-right: 5px;
```



```
margin-left: 5px;
```

```
}
```

```
.contri a{
```

```
text-decoration: none;
```

```
}
```

Company.html

```
<head>
```

```
<link rel="stylesheet" href="../static/style.css">
```

```
<link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
```

```
<link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Rancho&effect=shadow-multiple">
```

```
<title>Python Website Demo</title>
```

```
</head>
```

```
<body>
```

```
<br><br><br><br>
```

```
<center>
```

```
<h2>Browse all the Companies</h2><br>
```

```
<table>
```

```
<tr style="height: 60px; background-color: #04aec4; color: white;">
  <td style="visibility: hidden;"></td>
  <td>
    RECRUITER MAIL ID
  </td>
  <td>
    ORGANIZATION NAME
  </td>
  <td>
    ACTION
  </td>
</tr>
```

```
<tr>
  <form method="POST" action="/subscribe" name="form" onsubmit="required">
    <td><input type="text" name="candidateid" style="visibility: hidden;"></td>
    <td><input type="text" name="orgid" ></td>
    <td><input type="text" name="orgname"></td>
    <td>
      <button type="submit" name="submit" value="submit">
        <a href="#search" class="btn" type="button">Search</a>
      </button>
    </td>
  </form>
</tr>
```

```
</td>
```

```
</form>
```

```
</tr>
```

```
</table>
```

```
</frame>
```

```
</center>
```

```
<style type="text/css">
```

```
body{
```

```
background-color: white;
```

```
color: black;
```

```
}
```

```
table{
```

```
border-collapse: separate;
```

```
margin-left: -270px;
```

```
}
```

```
h2{
```

```
margin-right: 50px;
```

```
}
```

```
table tr{
```

```
height: 90px;
```

```
        background-color: white;

    }
table tr td{
    width: 200px;
    text-align: center;
}
#search{
    margin-left: -130px;
}
button{
    background-color: red;
    color: white;
    outline: none;
    border: none;
    cursor: pointer;
}
button:hover{
    opacity: 0.5;
}

</style>

<script type="text/javascript">
```

```
function search(){

}

</script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>


Contact.html

<!DOCTYPE html>

<html>

<head>

<title>JOB PORTAL CONTACT SERVICE </title>

</head>


<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet" type="text/css">

<link href="https://fonts.googleapis.com/css?family=Montserrat" rel="stylesheet" type="text/css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<style>

body{

background: linear-gradient(pink,white,rgb(207, 56, 139));
```

```
background-size: 500px;
overflow-x: hidden;

}

h1{
    text-align: center;
    color: rgb(226, 207, 209);
}

p{
    color: rgb(123, 176, 211);
}

.txt{
    text-align: center;
    margin-top: 245px;
    margin-left: 380px;
    margin-right: 250px;
    margin-bottom: 4px;
    height: 100px;
    width: 600px;
    border-color: black;
}

a {
text-decoration: none;
```

```
display: inline-block;
padding: 8px 16px;
}
.next {
background-color: #04AA6D;
text-align: start;
color: white;
}
a:hover {
background-color: #ddd;
color: black;
}
ul {
list-style-type: none;
margin: 0;
padding: 0;
}

li {
display: inline;
background:black;
}
.hdr{
background-color:black ;
```

```
width: 100%;
```

```
height: 70px;
```

```
}
```

```
a{
```

```
text-decoration: none;
```

```
}
```

```
.head{
```

```
background-image: linear-gradient(rgb(228, 105, 125),rgb(187, 118, 130));
```

```
width: 100%;
```

```
height: 150px;
```

```
}
```

```
</style>
```

```
<body >
```

```
<!--contact-->
```

```
<div class="head">
```

```
<h1 style="margin-top: 10px; margin-left:0.0px ;"> <b>JOB PORTAL HELP AND SERVICE</b></h1>
```

```
<h3 class="text-center" style="color: rgb(36, 24, 24);">Contact</h3>
```

```
<p class="text-center" style="color: rgb(22, 16, 16);"><em>Send your queries to us !</em></p>
```



</div>

<div id="contact" class="container" style="margin-top: 10px; margin-left:200px ;">

<div class="row">

<div class="col-md-4">

<p>Fan? Drop a note.</p>

<p><span class="glyphicon glyphicon-map-marker"></span>Chicago, US</p>

<p><span class="glyphicon glyphicon-phone"></span>Phone: +00 33473845775</p>

<p><span class="glyphicon glyphicon-envelope"></span>Email: abcmail@mail.com</p>

</div>

</div>

<div class="col-md-8">

<div class="row">

<div class="col-sm-6 form-group">

<input class="form-control" id="name" name="name" placeholder="Name" type="text" required>

</div>

<div class="col-sm-6 form-group">

<input class="form-control" id="email" name="email" placeholder="Email" type="email" required>

</div>

</div>

```
<textarea class="form-control" id="comments" name="comments" placeholder="Comment" rows="5"></textarea>
```

```
<br>
```

```
<div class="row">
```

```
<div class="col-md-12 form-group">
```

```
<form>
```

```
<button class="btn pull-right" type="submit">Send</button>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<br>
```

```
<ul class="nav nav-tabs">
```

```
</ul>
```

```
<div class="tab-content">
```

```
<div id="home" class="tab-pane fade in active">
```

```
<h2>About</h2>
```

```
<p style="color: rgb(12, 73, 12);">If you have any questions or concerns,don't hesitate to let me know and I look forward to hearing from you</p>
```

```
</div>
```

</div>

</body>

Sendgrid.py

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
```

```
body = ""
```

```
JOB RECOMMENDER
```

```
Hello first_name,
```

Thank you for creating a account in this App we hope to serve you the best. In the last years, job recommender systems have become popular since they successfully reduce information overload by generating personal-ized job suggestions. Although in the literature exists a variety of techniques and strategies used as part of job recommender systems, most of them fail to recommending job vacancies that fit properly to the job seekers profiles. Thus, the contributions of this work are threefold, we: i) made publicly available a new dataset formed by a set of job seekers profiles and a set

of job vacancies collected from different job search engine sites; ii) put forward the proposal of a framework for job recommendation based on professional skills of job seekers; and iii) carried out an evaluation to quantify empirically the recommendation abilities of two state-of-the-art methods, considering different configurations, within the proposed framework. We thus present a general panorama of job recommendation task aiming to facilitate research and real-world application design regarding this important issue.

Sincerely,

employee\_name

'''

```
sender = 'shruthiramesh2001@gmail.com'
```

```
password = 'kabchcujcuxrvdve'
```

```
receiver = 'shruthiramesh2001@gmail.com'
```

```
message = MIMEMultipart()
```

```
message['From'] = sender
```

```
message['To'] = receiver
```

```
message['Subject'] = 'This email has an attachment, a pdf file'
```

```
message.attach(MIMEText(body, 'plain'))
```

```
qimage = "img4.jpg"
```

```
binary_pdf = open(qimage, 'rb')
```

```
payload = MIMEBase('application', 'octate-stream', Name=qimage)
```

```
payload.set_payload((binary_pdf).read())
```

```
encoders.encode_base64(payload)
```

```
payload.add_header('Content-Decomposition', 'attachment', filename=qimage)
```

```
message.attach(payload)
```

```
session = smtplib.SMTP('smtp.gmail.com', 587)
```

```
session.starttls()
```

```
session.login(sender, password)
```

```
text = message.as_string()
```

```
session.sendmail(sender, receiver, text)
```

```
session.quit()
```

```
    print('Mail Sent')
```

```
app.js
```

```
const sign_in_btn = document.querySelector("#sign-in-btn");
const sign_up_btn = document.querySelector("#sign-up-btn");
const container = document.querySelector(".container");
```

```
sign_up_btn.addEventListener("click", () => {
  container.classList.add("sign-up-mode");
});
```

```
sign_in_btn.addEventListener("click", () => {
  container.classList.remove("sign-up-mode");
});
```

Db.py

```
import ibm_db
```

```
try:
    conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=2d46b6b4-cbf6-40eb-bbce-
6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=32328;SECURITY=SSL;SSlServerCertificate=DigiCertGlobalRootCA.
crt;UID=tzp28629;PWD=TVm24LZqNge08eTF;", "", "")
    print("db is connected")
except:
    print("db is not connected")
```

head.js

```
var text = ["Welcome to job Recommender", "Find a Startup Jobs", "Find a carrer you'll love", "Find Great places to work", "Find your dream job now", "Free Job Alerts", "Search by skill", "Search by company", "Search by job title"];
```

```
var counter = 0;
var elem = document.getElementById("changeText");
var inst = setInterval(change, 4000);
```

```
function change() {
  elem.innerHTML = text[counter];
  counter++;
  if (counter >= text.length) {
    counter = 0;

  }
}
```

Hide.js

```
setTimeout(() => {
  const box = document.getElementById('box');

  box.style.display = 'none';
}, 5000);
```

Reg.js

```
var text1 = ["On Registering,you can", "Build your profile", "Let recruiters find you","Get job postings delivered to your email","Find a job and grow your career"];
```

```
var counter1 = 0;
var elem1 = document.getElementById("changeText1");
var inst1 = setInterval(change1, 4000);
```

```
function change1() {
    elem1.innerHTML = text1[counter1];
    counter1++;
    if (counter1 >= text1.length) {
        counter1 = 0;
    }
}
```

Sign.js

```
function validateform(){
    var nameRegex = /^[a-zA-Z\-\-]+$/;
    var username = document.sign.username.value.match(nameRegex);
    var password=document.sign.password.value;
```

```
if(username == null){
    alert("Enter a Valid username");
```



```
    document.sign.username.focus();  
    return false;  
}  
else if(password.length<8){  
    alert("Password must be at least 8 characters long.");  
    return false;  
}  
}
```

GitHub & Project Demo Link

<https://github.com/IBM-EPBL/IBM-Project-10718-1659198612>

