

Skill/Job Recomender

1.INTRODUCTION:

1.1 Project Overview:

To develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations based on their skills. We can use a job search API to get the current job openings in the market which will fetch the data directly from the webpage.

1.2 Purpose:

In today world a lot of people have various skills but not a job for the skill. To overcome this problem, we have come up with a skill recommender solution through which the fresher or the skilled person can log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job.

2.Literatutre Survey:

LITERATURE SURVEY 1:

NAME OF THE PAPER : Job Recommendation based on Job Seeker Skills.

NAME OF THE AUTHOR :Jorge Valverde-Rebaza ,Ricardo Puma ,Paul Bustios,Nathalia C. Silva.

JOURNAL PUBLISHED : First Workshop on Narrative Extraction From Text co-located with 40th European Conference on Information Retrieval.

PUBLISHED MONTH : March

PUBLISHED YEAR 2018

OBJECTIVE OF THE PROJECT: ➤ In this ,when a candidate submits his/ her profile at a job seeker engine. ➤ Their job recommendations are mostly suggested taking their academic qualification and work experience into considerations.

LITERATURE SURVEY 2:

NAME OF THE PAPER : A survey of job recommender systems.

NAME OF THE AUTHOR : Shaha Alotaibi.

JOURNAL PUBLISHED : International Journal of Physical Sciences

PUBLISHED MONTH : July

PUBLISHED YEAR 2012

OBJECTIVE OF THE PROJECT: ➤ The recommender system technology aims to help users in finding items that match their personnel interests, it has a successful usage in e-commerce applications to deal with problems related to information overload efficiently. ➤ This article will present a survey of e-recruiting process and existing recommendation approaches for building personalized recommender systems for candidates/job.

LITERATURE SURVEY 3:

NAME OF THE PAPER : A Research of Job Recommendation System Based on Collaborative Filtering.

NAME OF THE AUTHOR : Cheng Yang, Yingya Zhang, Zhixiang Niu.

JOURNAL PUBLISHED : 2014 Seventh International Symposium on Computation Intelligence and Design.

PUBLISHED MONTH : December

PUBLISHED YEAR 2014

2.1 Existing Problem:

The major contribution of this work is as follows: The large MNC businesses use the mechanism currently in place for employment recommendations. The method is employed by businesses, not by regular people. If not, they will charge a small subscription fee to check the user's career options. The system functions for the average guy from city to village to modify this predicament. Because the students would look for employment based on their own skills, this approach will reduce unemployment. This company will also grow more quickly, which will result in more job openings. The goal of the proposed work is to suggest a job that is ideal for the user. It displays the hierarchical jobs that are best for the user, not just one job. Additionally, it suggests skills for the jobs that were suggested for the user. This project is intended for someone who simply has no idea what they are going to do.

2.2 References:

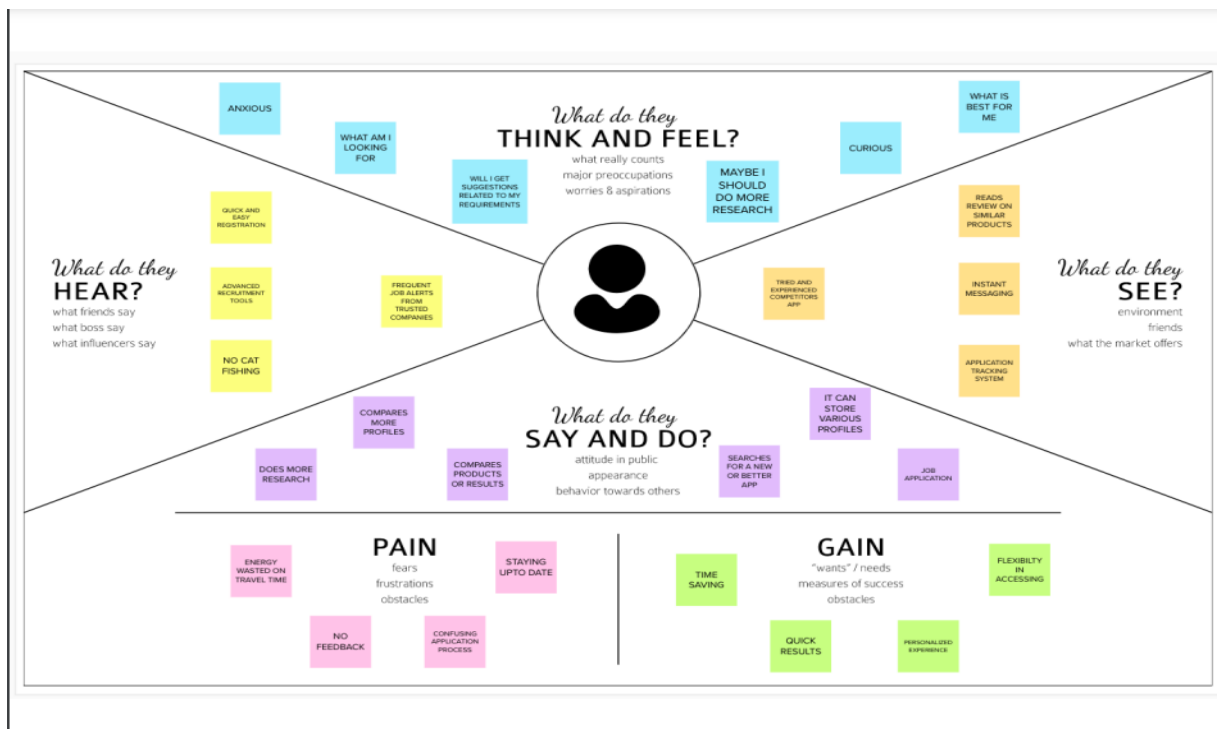
- <https://www.linkedin.com/pulse/writing-literature-review-research-experts-group/>
- https://www.researchgate.net/publication/356601605_Job_Recommender_Systems_A

2.3 Problem Statement Definition:

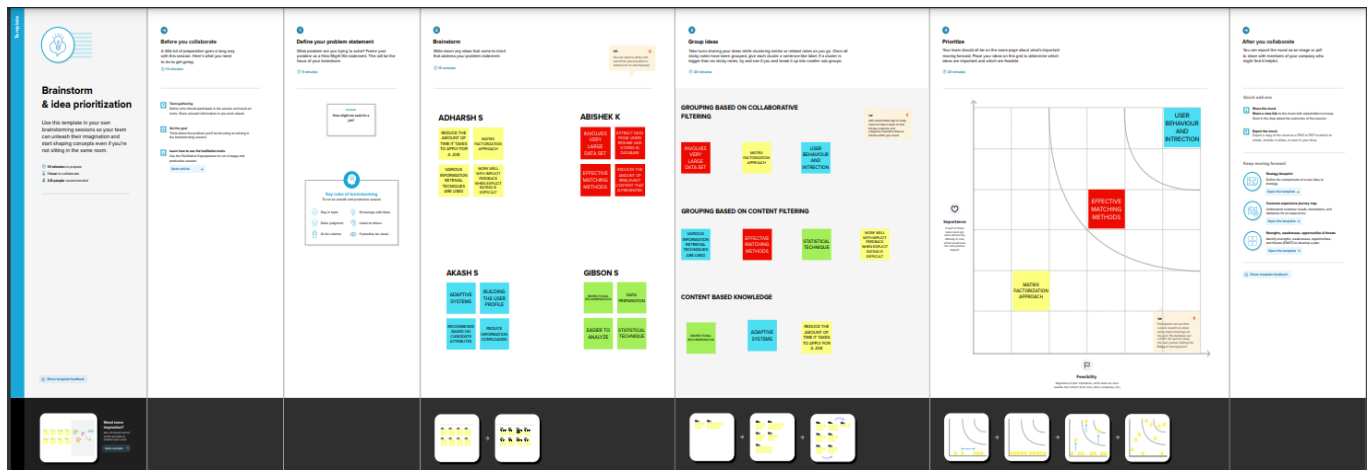
The problem statement aims at connecting the employer and job seekers where employers are the source of the resources and the job seeker can find and apply for their targeted job. It allows registered users to be able to search for jobs and filter the results based on required skills, salary, experience level, etc.

3.Ideation and Proposed Solution:

3.1 Empathy Map Canvas:



3.2 Ideation and Brainstorming:



3.3 Proposed Solution:

People who are having lot of skills and knowledge but not have enough opportunities to show their talents to get hired by reputed companies. We have come up with an app which provide our customers a lot of opportunities to show their talents to their recruiters which will ultimately leads to get hired by that recruiters' companies. It is different from online job searching websites by users are asked to feed skills along with education qualification into our app which help recruiters to filter the user's profile and hire them based on their skills. Getting hope by seeing the results and success stories of those who are already used our app got hired. Our app will provide life by providing our users' a good job. Our app needs an angel investors who are a wealthy private investors focused on financing small business ventures in exchange for equity. Our app also invests in companies who recruits skilled persons. Our app makes money through its talent solutions, marketing solutions, and premium subscriptions—in other words, by selling advertising, recruitment services, and membership privileges. Although not a primary revenue driver, its learning division, Learning Solutions, is also worth noting. Users can learn different skills and gain knowledge about different topics of interest. Our goal is connecting to your network for better job opportunities, so it will be scalable to a measurable extend.

3.4 Proposed Solution Fit:

Problem-Solution fit canvas 2.0

Purpose / Vision



4.Requirement Analysis :

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution:

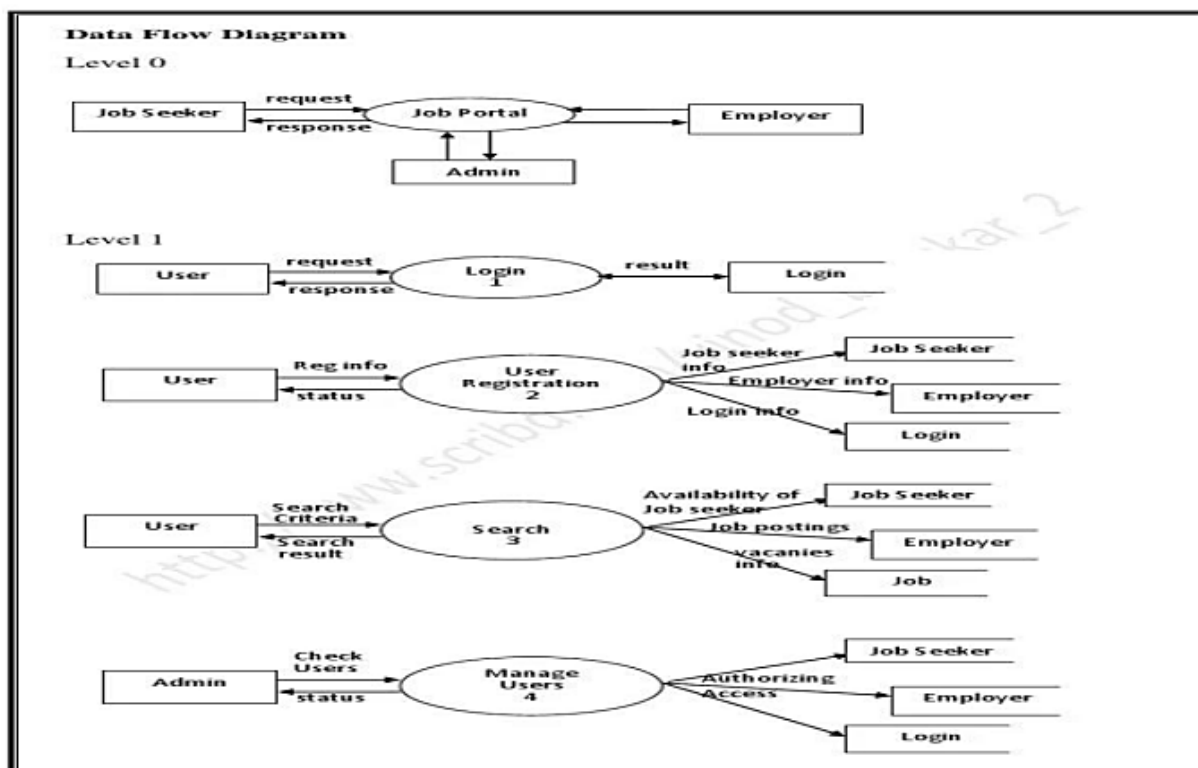
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail Registration through google form
FR-2	User Conformation	Confirmation via Email Confirmation via OTP
FR-3	User Login	User can login using login credentials – registered mail ID and 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:

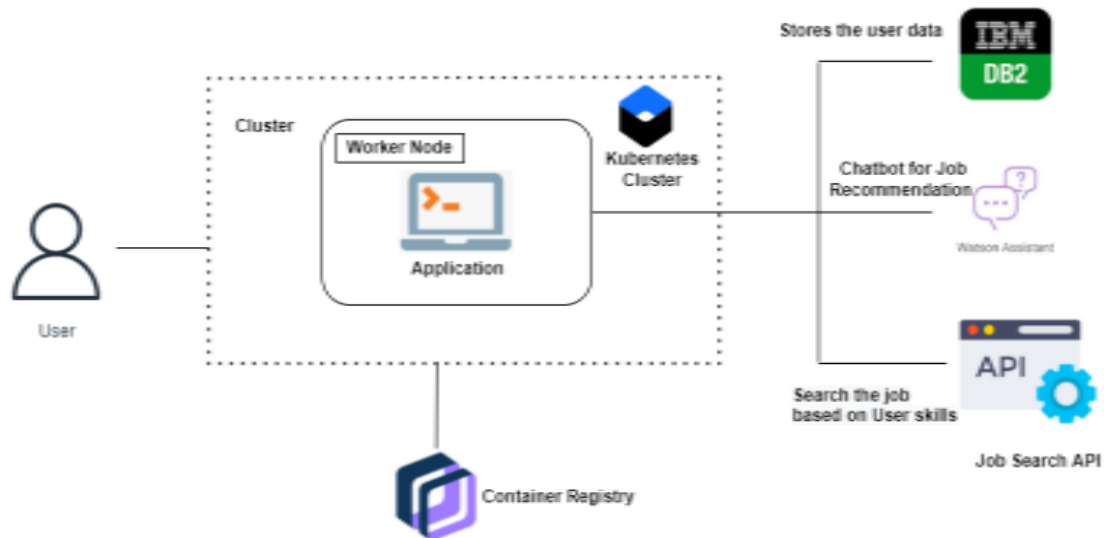
NFR No	Non Functional Requirement	Description
NFR-1	Usability	User can login to the application and can search the job based on the skillset and apply to the job. They can leave feedback to interview process and their experience
NFR-2	Security	Allow verified users to login Available security features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



5.3 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 Sign in 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 search for the job availability that matches with my job profile and criteria.	I can register & access the dashboard.	Low	Sprint-2
		USN-4	As a user, I would get the confirmation email from the registered company,		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard		The User can access the dashboard for further details.			
Customer (Web user)			The user can view the portal if there is any need.			
Customer Care Executive			The customer care executive clarifies the queries of the applicant.			
Administrator			The admin can view the list of applicants and access the entire portal.			

6. Project Planning and Scheduling:

6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement(Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	1	High	Adharsh S
Sprint-1			As a user, I will receive confirmation email once I have registered for the application	1	Medium	Adharsh S
Sprint-2	Login	USN-2	As a user, I can login the application by entering email & password	1	High	Abishek K
Sprint-2			As a user, I can search for job vacancies.	1	Medium	Abishek K
Sprint-3	Dashboard	USN-3	As a user, I can apply for jobs based on criteria	1	High	Akash S
Sprint-3			The applicants will be shortlisted based on job description.	1	Medium	Akash S
Sprint-4	Dashboard	USN-4	The candidates will be notified for the interview process.	1	High	Gibson S
Sprint-4			The selected candidates will be notified to their respective e-mail.	1	Medium	Gibson S

6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story points Completed	Sprint Release Date(Actual)
Sprint-1	20	6 days	24 Oct 2022	29 Oct 2022		29 Oct 2022

					20	
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

7.Coding & Solutioning:

7.1 Homepage:

This is the first page of the application.

```

1  <!DOCTYPE html>
2  <html lang="en" dir="ltr">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-
6          scale=1.0">
7      <link rel="stylesheet" href="../static/css/home.css">
8      <link rel="stylesheet"
9          href="https://cdnjs.cloudflare.com/ajax/libs/font-
10             awesome/5.15.2/css/all.min.css" />
11 </head>
12 <body>
13     <nav>
14         <div class="menu">
15             <div class="logo">
16                 <a href="/"> JOB RECOMMENDER WEBSITE </a>
17             </div>
18         </div>
19     </nav>
20 </body>
21 </html>

```

```

22         </div>
23
24     </nav>
25
26     <div class="img"></div>
27
28     <div class="center">
29
30         <div class="title">YOUR START TO FINAL DESTINATION</div>
31
32         <div class="sub_title">LAND ON YOUR DREAM JOB</div>
33
34         <div class="btns">
35
36             <button type="button"><a style="text-
37                 decoration:none;" href="signin.html">SIGN IN</a> </button>
38
39             <button type="button"><a style="text-
40                 decoration:none;" href="signup.html">SIGN UP</a> </button>
41
42         </div>
43
44     </div>
45
46 </body>
47
48 </html>

```

7.2 SignIn Page:

The user will be redirected to this page for sign in.

```

1  <!DOCTYPE html>
2
3  <html lang="en">
4
5  <head>
6      <title>JOB RECOMMENDER WEBSITE - SIGN IN</title>
7      <link rel="stylesheet" href="../static/css/style.css" />

```

```

8      <link
      href="https://fonts.googleapis.com/css2?family=Roboto:wght@300&di
      splay=swap" rel="stylesheet" />
9  </head>
10
11 <body>
12
13     <div class="login-box">
14
15         <h1 class="h1signin">SIGN IN</h1>
16
17         <form action="/user" method="post">
18
19             <label><b>EMAIL</b></label>
20
21             <input type="email" name="sem" placeholder="" />
22
23             <label><b>PASSWORD</b></label>
24
25             <input type="password" name="spw" placeholder="" />
26
27             <input type="submit" value="Submit" />
28
29         </form>
30
31     </div>
32
33     <p class="para-2">
34         DON'T HAVE AN ACCOUNT ? <a href="signup.html">SIGN UP</a>
      HERE
35     </p>
36
37 </body>
38
39 </html>

```

7.3 Sign Up Page:

The user will be redirected to this page if he is a first time user.

```
1  !DOCTYPE html>
2
3  <html lang="en">
4
5  <head>
6      <title>JOB RECOMMENDER WEBSITE - SIGN UP</title>
7      <link rel="stylesheet" href="../static/css/style.css" />
8      <link
9      href="https://fonts.googleapis.com/css2?family=Roboto:wght@300&di
10      splay=swap" rel="stylesheet" />
11 </head>
12
13 <body>
14
15     <div class="signup-box">
16
17         <h1 class="h1signup">SIGN UP</h1>
18
19         <form action="/app" method="post">
20
21             <label>FIRST NAME</label>
22
23             <input type="text" name="fn" placeholder="" />
24
25             <label>LAST NAME</label>
26
27             <input type="text" name="ln" placeholder="" />
28
29             <label>EMAIL</label>
30
31             <input type="email" name="em" placeholder="" />
32
33             <label>PASSWORD</label>
34
35             <input type="password" name="pw" placeholder="" />
36
37             <label>CONFIRM PASSWORD</label>
38
39             <input type="password" name="cpw" placeholder="" />
```

```

39         <input type="submit" value="Submit" />
40
41     </form>
42
43     <p>
44         By clicking the Sign Up button, you agree to our <br
45     />
46         <a href="#">Terms and Condition</a> and <a
47     href="#">Policy Privacy</a>
48     </p>
49 </div>
50
51 <p class="para-1">
52     ALREADY HAVE AN ACCOUNT ? <a href="signin.html">SIGN
53     IN</a> HERE
54 </p>
55 </body>
56
57 </html>

```

7.4 Welcome Page:

This page welcomes the new user and redirects the user to SignIn page.

```

1 <!DOCTYPE html>
2
3 <html lang="en" dir="ltr">
4
5 <head>
6     <meta charset="UTF-8">
7     <meta name="viewport" content="width=device-width, initial-
8     scale=1.0">
9     <link rel="stylesheet" href="../static/css/home.css">
10    <link rel="stylesheet"
11    href="https://cdnjs.cloudflare.com/ajax/libs/font-
12    awesome/5.15.2/css/all.min.css" />

```

```
10 </head>
11
12 <body>
13
14     <nav>
15
16         <div class="menu">
17
18             <div class="logo">
19
20                 <a href="/"> JOB RECOMMENDER WEBSITE </a>
21
22             </div>
23
24         </div>
25
26     </nav>
27
28     <div class="img"></div>
29
30     <div class="center">
31
32         <div class="title">SUCCESSFULLY REGISTERED</div>
33
34         <div class="sub_title">SIGN IN WITH THE RECEIVED
35 CREDENTIALS</div>
36
37         <div class="btns">
38
39             <button type="button"><a style="text-
40 decoration:none;" href="signin.html">SIGN IN</a> </button>
41
42         </div>
43
44 </body>
45
46 </html>
```

7.5 User.html :

The user enters the skill in this page :

```
1  <!DOCTYPE html>
2
3  <html lang="en" dir="ltr">
4
5  <head>
6      <meta charset="UTF-8">
7      <meta name="viewport" content="width=device-width, initial-
      scale=1.0">
8      <link rel="stylesheet" href="../static/css/user.css">
9      <link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
      awesome/5.15.2/css/all.min.css" />
10 </head>
11
12 <body>
13
14     <section>
15
16         <nav>
17
18             <div class="menu">
19
20                 <div class="logo">
21
22                     <a href="/"> JOB RECOMMENDER WEBSITE </a>
23
24                 </div>
25
26             </div>
27
28         </nav>
29
30         <div class="img"></div>
31
32         <div class="center">
33
34             <div class="title">PLEASE ENTER YOUR JOB SKILL</div>
```

```

35
36         <div class="btns">
37
38             <form action="/job" method="post">
39
40                 <input type="text" name="search"
class="search">
41
42                 <input type="submit" value="submit">
43
44             </form>
45
46         </div>
47
48     </div>
49
50 </section>
51
52 </body>
53
54 </html>

```

7.6 Docker:

This code is used to connect the app to docker and flask:

```

1 FROM python:3.8
2 MAINTAINER Abishek Kumar <anjaanabishek10@gmail.com>
3 WORKDIR /app
4 COPY . /app
5 RUN pip install -r requirements.txt
6 EXPOSE 5000
7 CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]

```

7.7 App.py:

This is the main python code for the application:

```

1 import re
2 import os

```



```
3 import ibm_db
4 from flask import flash
5 from flask import Flask
6 from flask import request
7 from flask import redirect
8 from bs4 import BeautifulSoup
9 from flask import render_template
10 from urllib.request import urlopen
11 from urllib.request import Request
12 from sendgrid import SendGridAPIClient
13 from sendgrid.helpers.mail import Mail
14
15 app = Flask(__name__)
16
17 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-
    46ec-a1fc-
    1c999edb6187.c3n41cmd0nqn timerk39u98g.databases.appdomain.cloud;PORT=
    31249;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;
    UID=txz44264;PWD=0e7ZwhLpS1dJFh9f","", "")
18
19 @app.route("/")
20
21 def home():
22
23     return render_template("/homepage.html")
24
25 @app.route("/signup.html")
26
27 def signup():
28
29     return render_template("/signup.html")
30
31 @app.route("/signin.html")
32
33 def signin():
34
35     return render_template("/signin.html")
36
37 @app.route("/user", methods=["POST", "GET"])
38
```

```
39 def user():
40
41     if (request.method == "POST"):
42
43         sem = request.form.get("sem")
44         spw = request.form.get("spw")
45         em_lst = []
46         pw_lst = []
47
48         try:
49
50             data = ibm_db.exec_immediate(conn, "SELECT \"EMAIL\",
51             \"PASSWORD\" FROM \"TXZ44264\".\"SIGNUPUSERDETAILS\";")
52
53             while ibm_db.fetch_row(data) != False:
54
55                 em_lst.append(ibm_db.result(data, 0).replace(" ",
56                 ""))
57                 pw_lst.append(ibm_db.result(data, 1).replace(" ",
58                 ""))
59
60                 if (sem in em_lst and spw in pw_lst):
61
62                     return render_template("/user.html")
63
64                 else:
65
66                     return redirect("/signin.html")
67
68             except Exception as e:
69
70                 return render_template("/error.html")
71
72 @app.route("/app", methods=["POST", "GET"])
73
74 def welcome():
75
76     if (request.method == "POST"):
77
78         fn = request.form.get("fn")
```

```

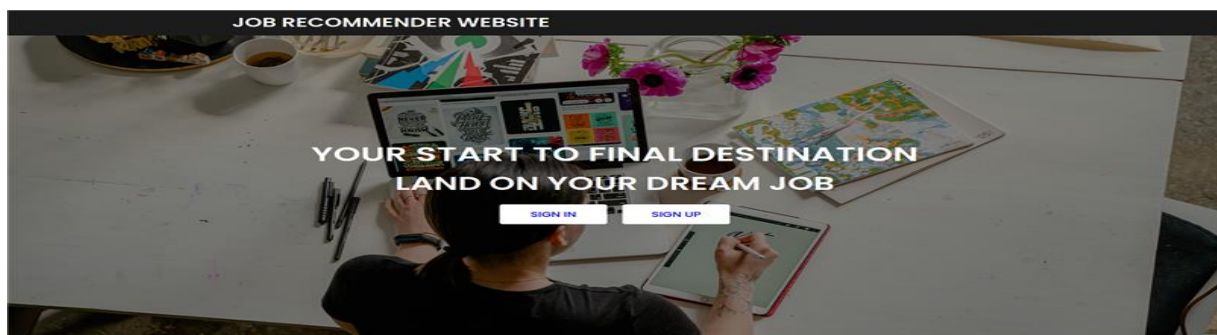
76         ln = request.form.get("ln")
77         em = request.form.get("em")
78         pw = request.form.get("pw")
79         cpw = request.form.get("cpw")
80
81         message = Mail(
82             from_email = "312819104004@act.edu.in",
83             to_emails = em,
84             subject = "Job Recommender - Account created
successfully",
85             html_content = '<p>You can now sign in with your
credentials,<br>email: {}<br>password: {}</p>'.format(em, pw)
86         )
87
88         try:
89
90             insert = ibm_db.exec_immediate(conn, "INSERT INTO
\"TXZ44264\\\".\"SIGNUPUSERDETAILS\" VALUES ('{}', '{}', '{}',
'{}');".format(fn, ln, em, pw))
91             f = open("./sendgrid.txt", "r")
92             sg = SendGridAPIClient(f.read())
93             response = sg.send(message)
94
95             return render_template("/welcome.html")
96
97         except Exception as e:
98
99             print(e)
100
101             return render_template("/error.html")
102
103 @app.route("/job", methods=["POST", "GET"])
104
105 def job():
106
107     if (request.method == "POST"):
108
109         search = request.form.get("search")
110         link = "https://www.glassdoor.com/Job/india-{}-jobs-
SRCH_IL.0,5_IN115_K06,11.htm".format(search)

```

```
111
112     return render_template("/job.html", link=link)
113
114 if __name__ == "__main__":
115     app.run(host = "0.0.0.0", port = 5000, debug = True)
```

8.Results:

8.1 Homepage:



8.2 Sign In :



8.3 Sign Up :

SIGN UP

FIRST NAME

LAST NAME

EMAIL

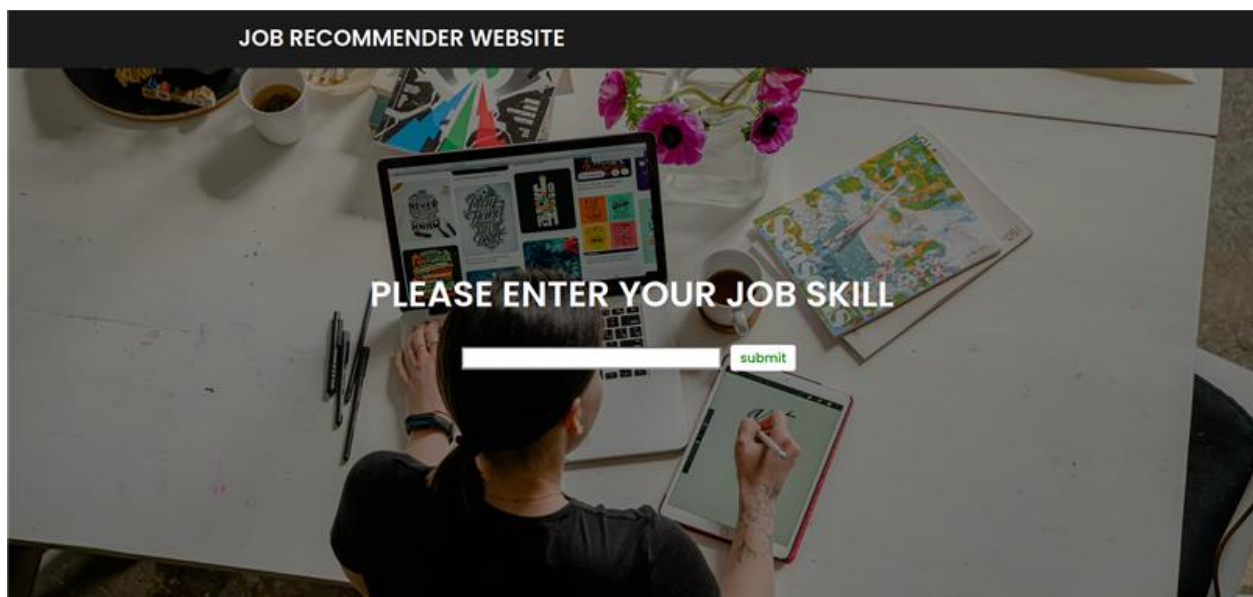
PASSWORD

CONFIRM PASSWORD

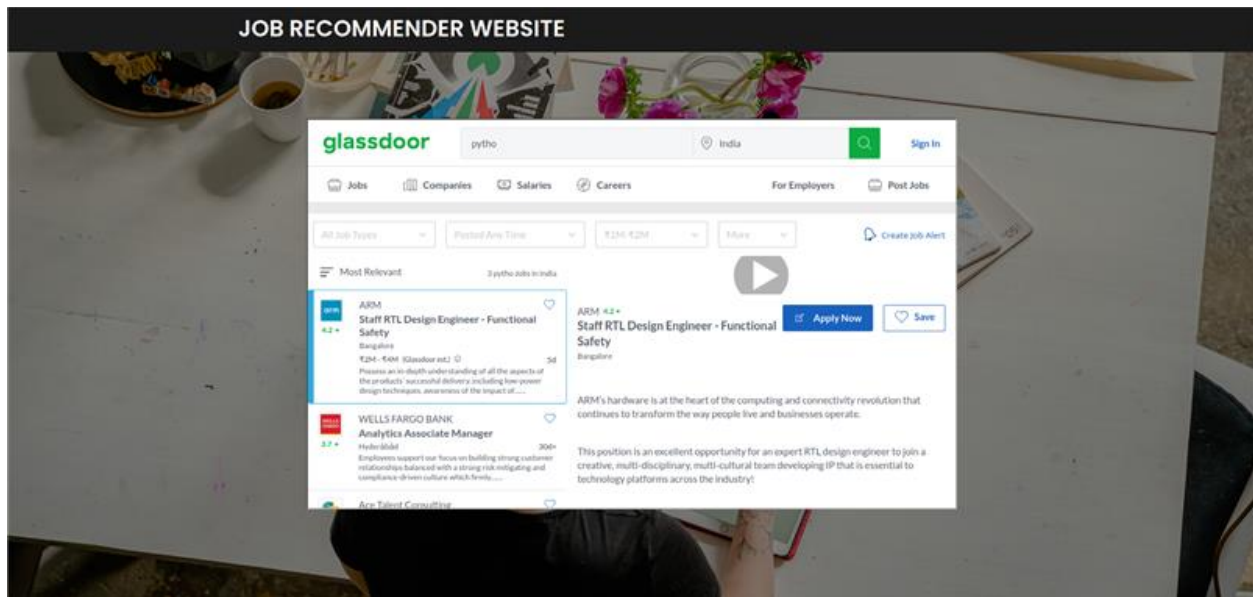
By clicking the Sign Up button, you agree to our Terms and Conditions and Privacy Policy

ALREADY HAVE AN ACCOUNT? [CLICK HERE](#)

8.4 Search Page :



8.5 Job Page :



9.Conclusion:

This an Job/Skill Recomender Application which gets the skill from the user and searches for the perfect job for the skill using an third party application Glassdoor.

Github Link:

<https://github.com/IBM-EPBL/IBM-Project-35498-1660285278>