

PROJECT REPORT

IBM-Project-23327- 1659878210

TITLE:

Skill / Job Recommender Application

TEAM LEADER:

N ANBARAN ARIVUKOE

TEAM MEMBER:

- A ARAVINDA KRISHNAN
- J TAVAMANI RAJA DURAI
- N S SANTHOSH SIVAN

TEAM ID:

PNT2022TMID22792

TABLE OF CONTENTS	PAGE NO
1. INTRODUCTION 1.1 Project Overview 1.2 Purpose	1-2
2. LITERATURE SURVEY 2.1 Existing problem 2.2 References 2.3 Problem Statement Definition	3-5
3. IDEATION & PROPOSED SOLUTION 3.1 Empathy Map Canvas 3.2 Ideation & Brainstorming 3.3 Proposed Solution 3.4 Problem Solution fit	6-9
4. REQUIREMENT ANALYSIS 4.1 Functional requirement 4.2 Non-Functional requirements	9-10
5. PROJECT DESIGN 5.1 Data Flow Diagrams 5.2 Solution & Technical Architecture 5.3 User Stories	10-13
6. PROJECT PLANNING & SCHEDULING 6.1 Sprint Planning & Estimation 6.2 Sprint Delivery Schedule 6.3 Reports from JIRA	14-15
7. CODING & SOLUTIONING 7.1 Feature 1 7.2 Feature 2	15-16
8. TESTING 8.1 Test Cases 8.2 User Acceptance Testing	17

9. RESULTS 9.1 Performance Metrics	18
10. ADVANTAGES & DISADVANTAGES	19-20
11. CONCLUSION	20
12.FUTURE SCOPE	21
13. APPENDIX	22

1 INTRODUCTION

1.1 Project Overview

Job recommendation is an important task for the modern recruitment industry. An excellent job recommender system not only enables to recommend a higher paying job which is maximally aligned with the skill-set of the current job, but also suggests to acquire few additional skills which are required to assume the new position.

In this work, we created three types of information networks from the historical job data: (i) job transition network, (ii) job-skill network, and (iii) skill co-occurrence network. We provide a representation learning model which can utilize the information from all three networks to jointly learn the representation of the jobs and skills in the shared k -dimensional latent space.

In our experiments, we show that by jointly learning the representation for the jobs and skills, our model provides better recommendation for both jobs and skills. Additionally, we also show some case studies which validate our claim.

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.

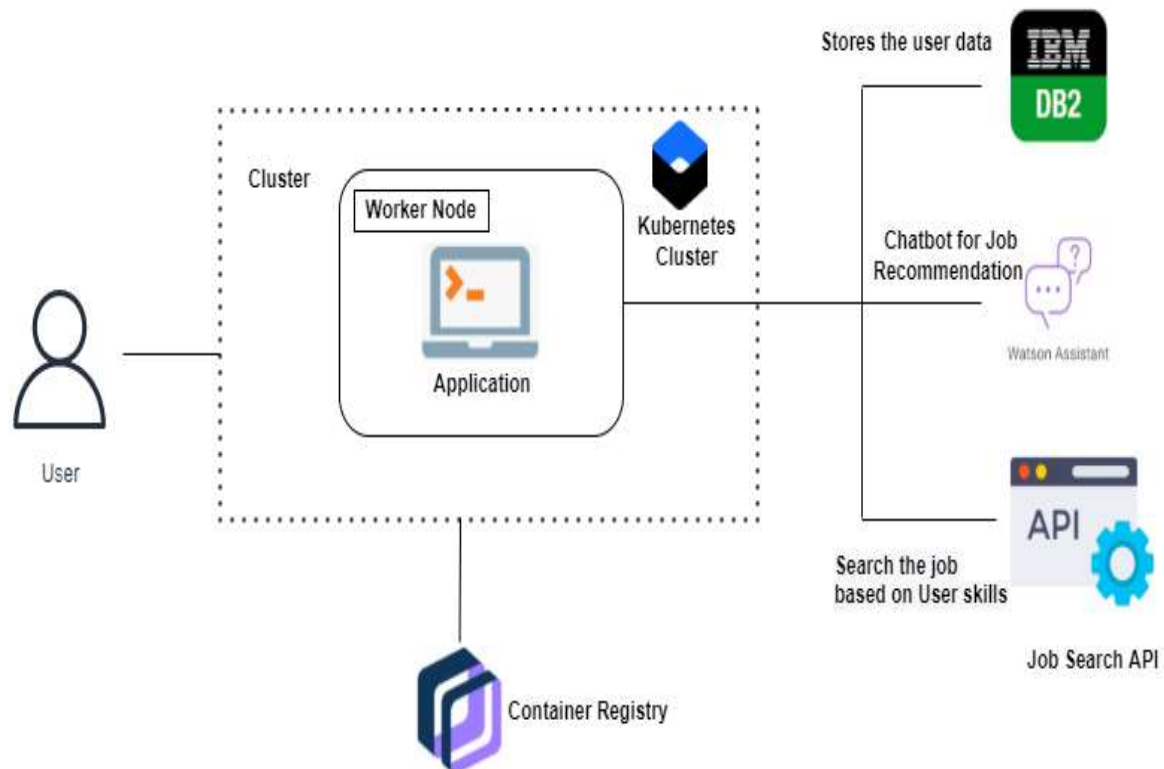
Users will interact with the chatbot and can get the recommendations based on their skills.

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

An excellent job recommender system not only enables to recommend a higher paying job which is maximally aligned with the skill-set of the current job, but also suggests to acquire few additional skills which are required to assume the new position.

To create a software that filters the job based on the skillset of the candidates who are seeking for the job. Then that filtered job is recommended for that candidates based on their skillset.



2 LITERATURE SURVEY

2.1 Existing problem

“Job Recommendation based on Job Seeker Skills. Jorge ValverdeRebaza ,Ricardo Puma ,Paul Bustios ,Nathalia C. Silva. : First Workshop on Narrative Extraction From Text co-located with 40th European Conference on Information Retrieval March 2018”

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.

A survey of job recommender systems, Shaha Alotaibi ,International Jounal of Physical Sciences July 2012”

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 matching

A Research of Job Recommendation System Based on Collaborative Filtering: Cheng Yang,Yingya Zhang,Zhixiang Niu, 2014 Seventh International Symposium on Computational Intelligence and Design, December 2014”

It analyze the candidate’s resume and the companies’ recruitment guidelines.

To compare and come to a better conclusion upon finding the best suited candidates for the job

2.2 References

[1] Schafer J B, Frankowski D, Herlocker J, et al.

Collaborative filtering recommender systems[M]//The adaptive web. Springer Berlin Heidelberg, 2007: 291-324.

[2] Pazzani M J, Billsus D. Content-based recommendation

systems[M]//The adaptive web. Springer Berlin Heidelberg, 2007:325-341.

[3] Sarwar B, Karypis G, Konstan J, et al. Item-based

collaborative filtering recommendation

algorithms[C]//Proceedings of the 10th international conference on World Wide Web. ACM, 2001: 285-295.

[4] Nikolaos D. Almalis ,Prof. George A. Tsihrintzis ,

Nikolaos Karagiannis et al. "FoDRA - A New Content-Based Job Recommendation Algorithm for Job Seeking and Recruiting".

[5] Anika Gupta, Dr. Deepak Garg. "Applying Data Mining

Techniques in Job Recommender System for Considering Candidate Job Preferences".

[6] Dunning T. Accurate methods for the statistics of

surprise and coincidence[J]. Computational linguistics, 1993, 19(1): 61-74.

[7] Emmanuel Malherbe , Mamadou Diaby , Mario Cataldi

et al. "Field Selection for Job Categorization and Recommendation to Social Network Users". 2014

IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014).

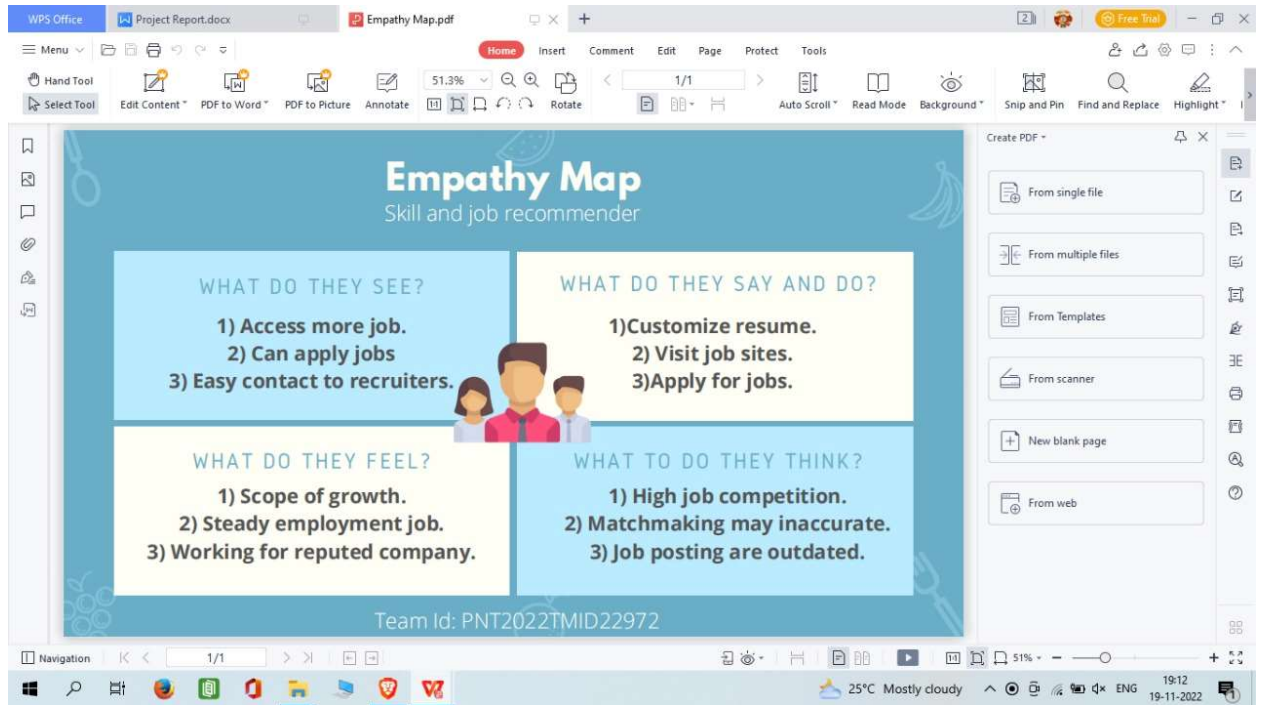
2.3 Problem Statement Definition

Having lots of skills but wondering which job will best suit you? Don't need to worry! 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

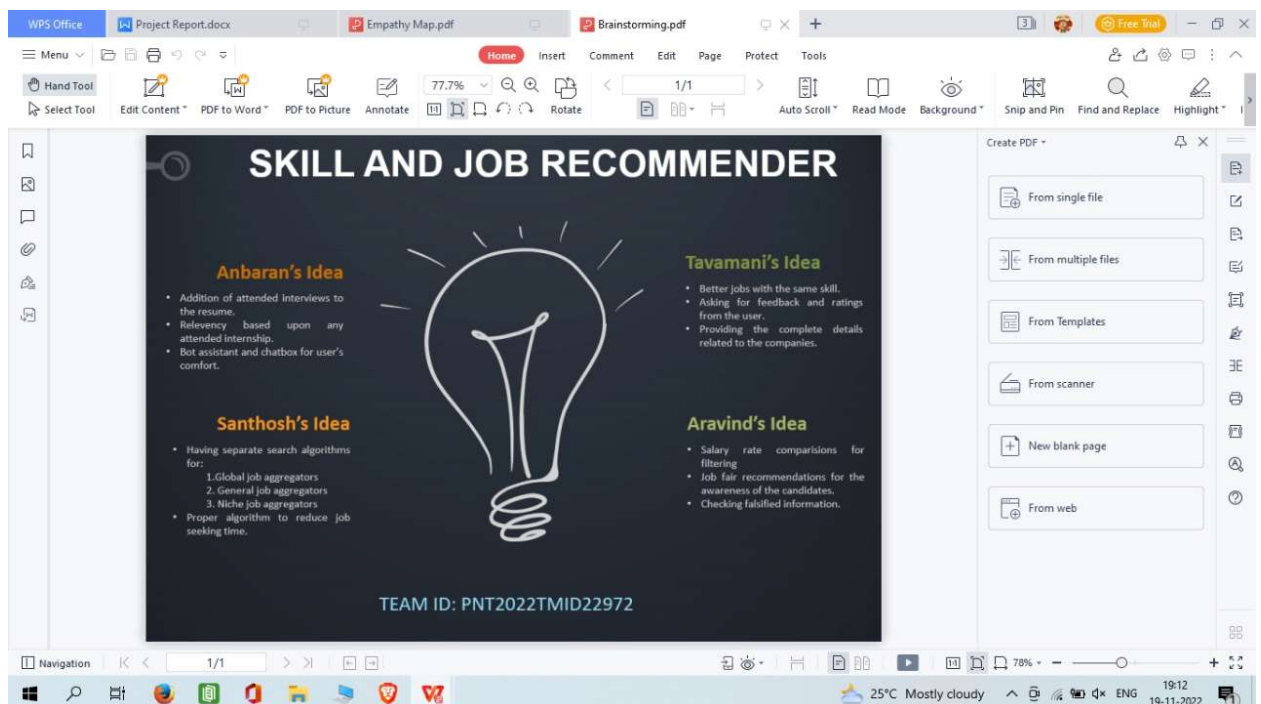
To develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations 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

3 IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<p>Having lots of skills but wondering which job will best suit you? Don't need to worry! 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.</p> <p>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.</p>

2.	Idea / Solution description	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 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.
3.	Novelty / Uniqueness	The best position are suggested to any person according to her skills. While the position of known profiles are assumed

3.4 Problem Solution fit

Problem-Solution fit canvas 2.0 Skill/Job recommender Team ID - PNT2022TMID23025

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? i.e. working parents of 0-5 y.o. kids The main customers for our project are : <ul style="list-style-type: none"> Persons who are seeking employment Persons that recruit job candidates 	6. CUSTOMER CONSTRAINTS CC What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices. <ul style="list-style-type: none"> Concern about misuse of personal information Worry about unreliable connections Inadequate product knowledge Potential Scam Time consuming 	5. AVAILABLE SOLUTIONS AS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking <table border="1"> <tr> <th>Pros</th> <th>Cons</th> </tr> <tr> <td>Promotion of people's skillset</td> <td>Delivering false information</td> </tr> <tr> <td>Marketing of company infrastructure</td> <td>Occurrence of fraudulent activity</td> </tr> <tr> <td>Cultivate commercial relationship</td> <td>Intense competition</td> </tr> </table>	Pros	Cons	Promotion of people's skillset	Delivering false information	Marketing of company infrastructure	Occurrence of fraudulent activity	Cultivate commercial relationship	Intense competition	Explore AS, differentiate
	Pros	Cons										
Promotion of people's skillset	Delivering false information											
Marketing of company infrastructure	Occurrence of fraudulent activity											
Cultivate commercial relationship	Intense competition											
2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <ul style="list-style-type: none"> Create a platform to facilitate job searching A platform to make it simpler to identify people with the necessary skills Make the job-filtering process simpler Profile with safe personal data 	9. PROBLEM ROOT CAUSE RC What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations. <ul style="list-style-type: none"> Jobs that are listed on unreliable platforms may be fraudulent Companies fail to disclose their true infrastructure Some job portals want payment in advance of the job starting. Users post false credentials Users pretend to have expertise in a skillset they lack 	7. BEHAVIOUR BE What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace) <ul style="list-style-type: none"> When Users apply for fraudulent jobs, they get unhappy due to wasted time Users were not satisfied when platforms allowed hirers to post jobs that were not real Cheating during online recruitment process When candidates with inadequate qualifications apply for a position, employers become irritated. 	Focus on J&P, tap into C									
3. TRIGGERS TR What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. <ul style="list-style-type: none"> Employment opportunities Branding Endorsement and connections Get job alerts 	10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. To develop an end-to-end web application which in default have a lot of current job openings through job search API out of which appropriate job will be recommended based on user skill set. At the same time students can develop their skills side by side with various courses and webinars offered by reputed organization. In addition to this a smart chat bot will be available for 24*7 which can help users in finding the right job.	8. CHANNELS OF BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 <ul style="list-style-type: none"> Apply for jobs Review job applications Attend initial level assessment 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. <ul style="list-style-type: none"> Final level interview Checkout location and infrastructure of company Finalize paperwork 		Extract online & offline CH of BE								
4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? <table border="1"> <tr> <th>Emotions-Before</th> <th>Emotions-After</th> </tr> <tr> <td>Lack of knowledge about job vacancy</td> <td>User receive updates on job vacancies.</td> </tr> <tr> <td>No proper platform to showcase skillset</td> <td>Exhibit skillset in profile</td> </tr> <tr> <td>More paperwork during recruitment</td> <td>Easy recruitment process</td> </tr> </table>	Emotions-Before	Emotions-After	Lack of knowledge about job vacancy		User receive updates on job vacancies.	No proper platform to showcase skillset	Exhibit skillset in profile	More paperwork during recruitment	Easy recruitment process			
Emotions-Before	Emotions-After											
Lack of knowledge about job vacancy	User receive updates on job vacancies.											
No proper platform to showcase skillset	Exhibit skillset in profile											
More paperwork during recruitment	Easy recruitment process											

4 REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Sign in / Login	Register with username, password
FR-2	Profile Registration	Register with username, password, email, qualification, skills. This data will be stored in a database.
FR-3	Job profile display	Display job profiles based on availability, location, skills.
FR-4	Chatbot	A chat on the webpage to solve user queries and issues.
FR-5	Job Registration	The company's registration/Description details will be sent to the registered email id of the user.
FR-6	Logout	Use logout option after completing job registration process.

4.2 Non-Functional requirements

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

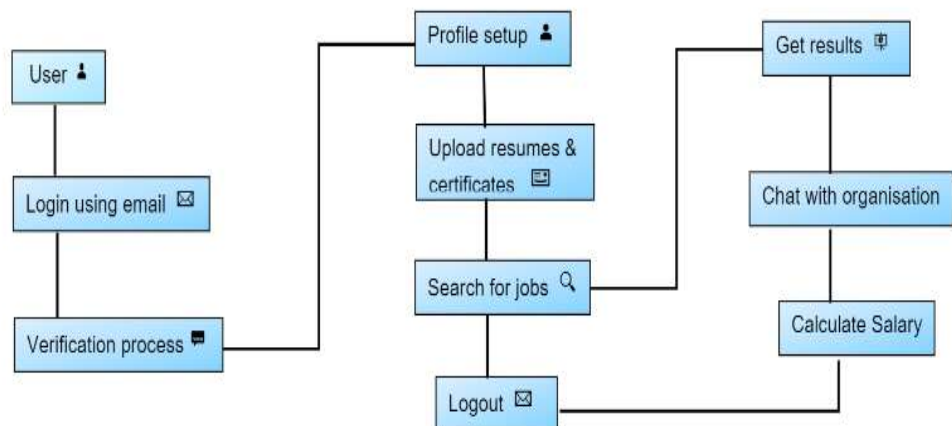
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The webpage will be designed in such a way that any non-technical user can easily navigate through it and complete the job registration work. (easy and simple design)
NFR-2	Security	Using of python flask to cloud connect will provide security to the project. Database will be safely stored in DB2.
NFR-3	Reliability	To make sure the webpage doesn't go down due to network traffic.
NFR-4	Performance	Focus on loading the webpage as quickly as possible irrespective of the number of user/integrator traffic.
NFR-5	Availability	The webpage will be available to all users (network connectivity is necessary) at any given point of time.
NFR-6	Scalability	Increasing the storage space of database can increase the number of users. Add some features in future to make the webpage unique and attractive.

5 PROJECT DESIGN

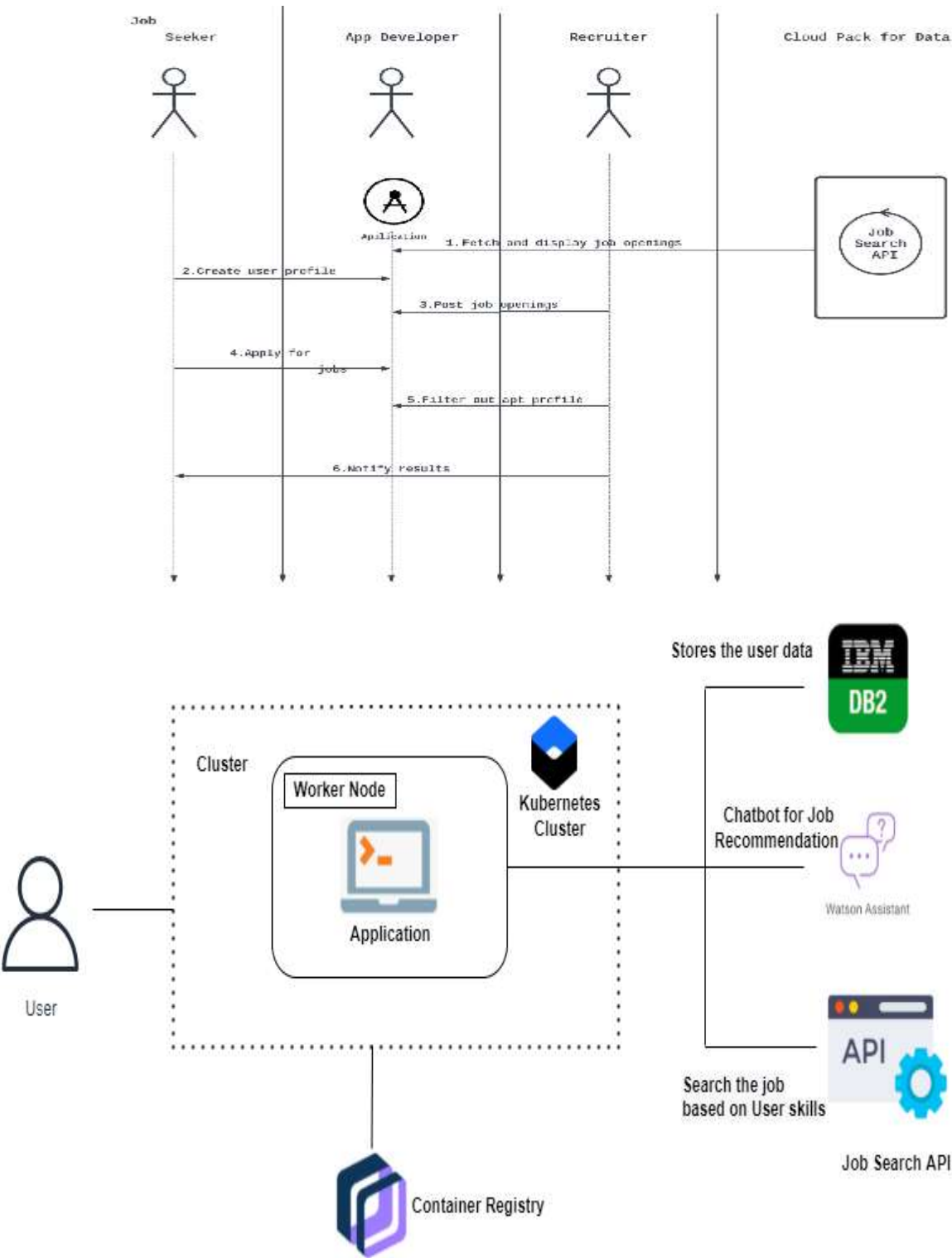
5.1 Data Flow Diagrams

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture



5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

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		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-5	As a user, I can access my dashboard after signing in.	I can access my account / dashboard	High	Sprint-1
Customer (Web user)	Access	USN-6	As a user, I can setup a profile, and basic details by signing in.			
		USN-7	As a user, I will upload my resume, certificates, and other requirements.	I can perform several task in the application	Medium	Sprint-1
Customer Care Executive	Chatbot	USN-8	As a user, I can seek guidance from the customer care executive.		High	Sprint-1
Administrator	DBMS	USN-9	As a administrator, I can keep the applications of your organization relies on running.	I can perform various modifications in the applications.	High	Sprint-1

6 PROJECT PLANNING & SCHEDULING

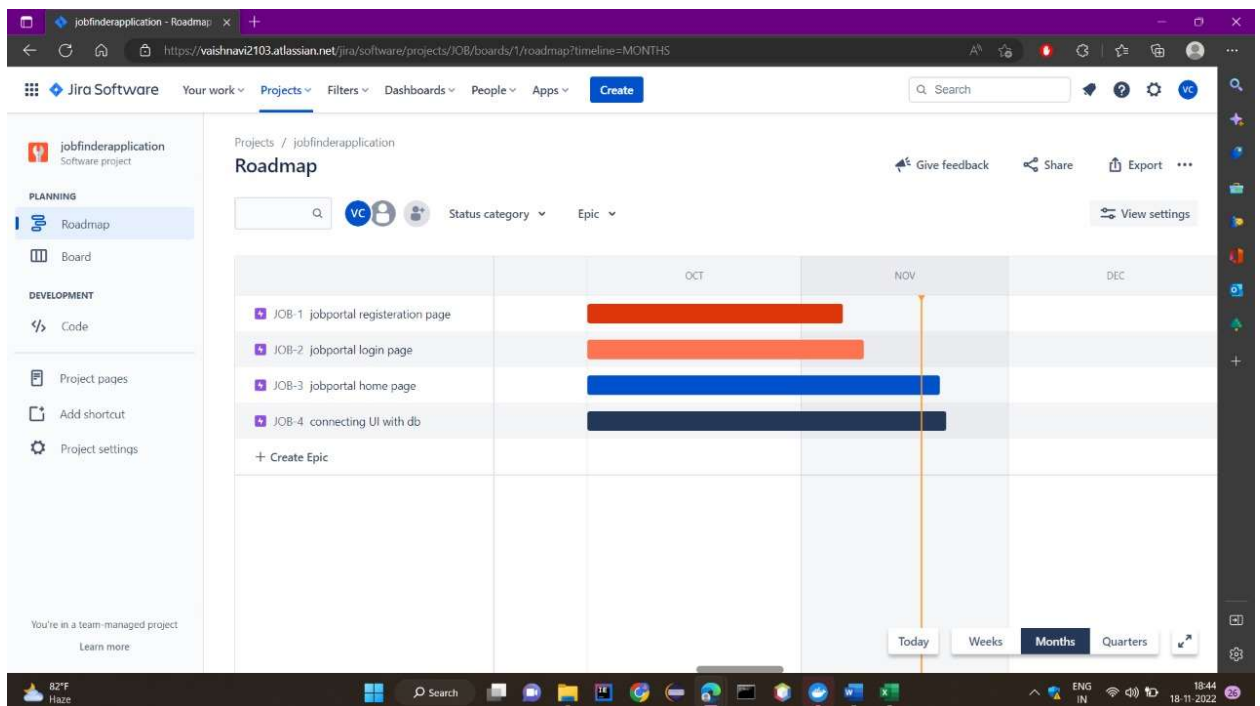
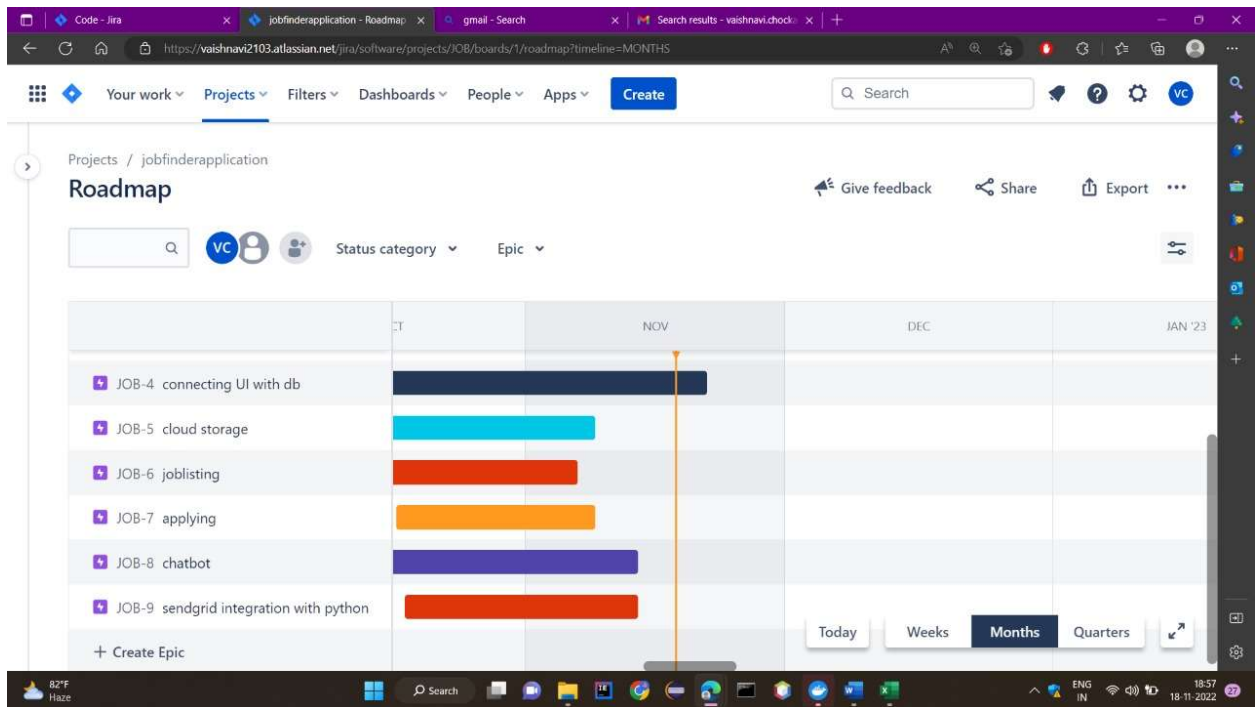
6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	UI Creation Creating Registration page, Login page	10	Medium	J YAVAMANI RAJADURAI N ANBARANI ARIVUKOE A ARAVINDA KRISHNAN N S SANTHOSH SIVAN
Sprint-1	Database Connectivity	USN-2	Viewing and applying jobs Connecting UI with Database	10	High	N ANBARANI ARIVUKOE A ARAVINDA KRISHNAN
Sprint-2	SendGrid Integration	USN-3	SendGrid Integration with Python Code	10	Low	J YAVAMANI RAJADURAI N S SANTHOSH SIVAN
Sprint-2	Chatbot Development	USN-4	Building a chatbot	10	High	N ANBARANI ARIVUKOE A ARAVINDA KRISHNAN
Sprint-3	Integration and Containerisation	USN-5	Integrating chatbot to the HTML page and containerizing the app.	20	Medium	J YAVAMANI RAJADURAI N ANBARANI ARIVUKOE A ARAVINDA KRISHNAN N S SANTHOSH SIVAN
Sprint-4	Upload image and deployment	USN-6	Upload the image to the IBM Registry and deploy it in the Kubernetes Cluster.	20	High	J YAVAMANI RAJADURAI N S SANTHOSH SIVAN

6.2 Sprint Delivery Schedule

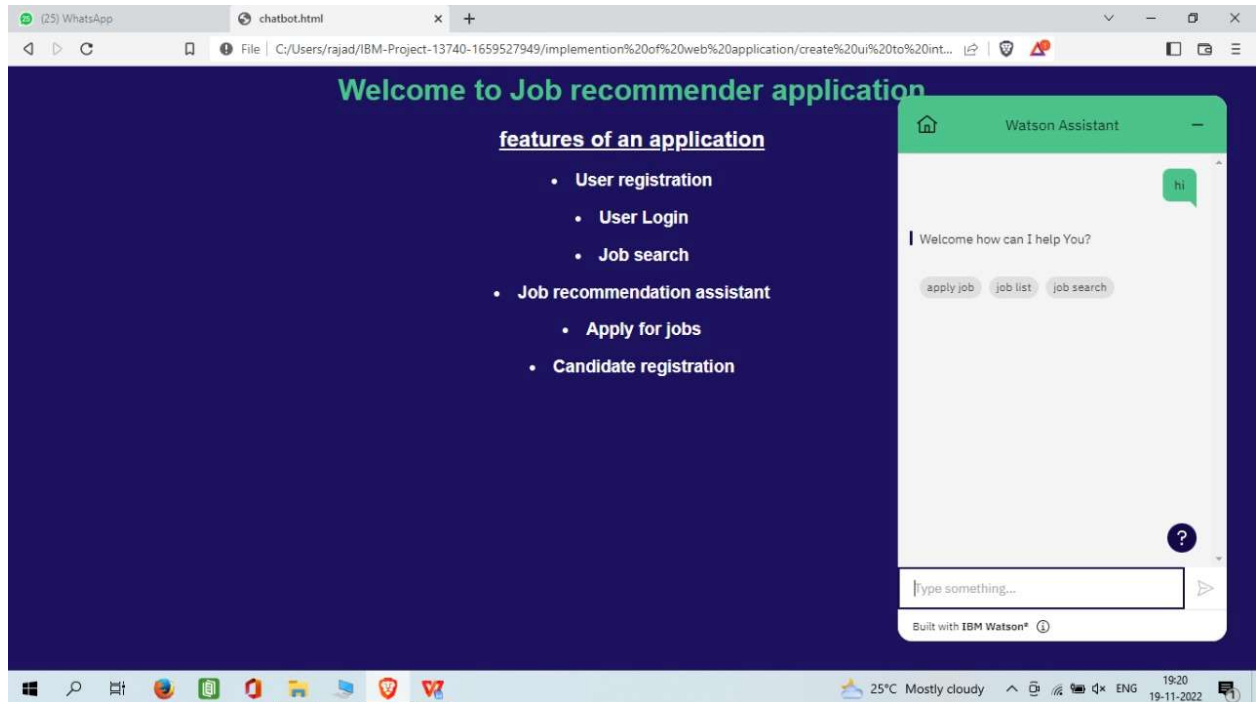
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	18	6 Days	24 Oct 2022	29 Oct 2022	18	29 Oct 2022
Sprint-2	27	6 Days	31 Oct 2022	05 Nov 2022	27	05 Nov 2022
Sprint-3	29	6 Days	07 Nov 2022	12 Nov 2022	29	12 Nov 2022
Sprint-4	14	6 Days	14 Nov 2022	19 Nov 2022	14	19 Nov 2022

6.3 Reports from JIRA



7 CODING & SOLUTIONING

7.1 Feature 1



The software has an In-built “Chat Bot” which can help assist with ongoing queries and provide fast and effective solutions to user problems which may occur and also redirect to management attention if need be there any complications the customer service will be available 24*7 to assist in case of any controversial issues arise

7.2 Feature 2

In this project we have created the dashboard page to view the jobs available and to make ease to access the website

- They communicate information quickly.
- They display information clearly and efficiently.
- They show trends and changes in data over time.
- They are easily customizable.
- The most important widgets and data components are effectively presented in a limited space.

The screenshot shows a web browser window with a single tab titled 'APPLY JOBS'. The address bar displays a file path: 'C:/Users/rajad/IBM-Project-13740-1659527949/implementation%20of%20web%20application/create%20ui%20to%20int...'. The main content area has a dark blue background with the title 'APPLY JOBS' in green. Below the title, there are six white input fields, each preceded by a label in green: 'ID', 'NAME', 'SKILLS', 'GENDER', 'EMAIL', and 'PHONE'. The Windows taskbar at the bottom shows various icons, including the Start button, search, and several application icons. The system tray on the right indicates a temperature of 25°C, 'Mostly cloudy' weather, and the time 19:20 on 19-11-2022.

8 TESTING

8.1 Test Cases

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

This Software is tested and evaluated successfully.

8.2 User Acceptance Testing

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Inventory Management System project at the time of the release to User Acceptance Testing (UAT)

User Acceptance Testing is carried out in a separate testing environment. A change, an update, or a new feature is requested and developed. Unit and integration tests are run. All seems to be in order. But then, after it is released to the public, serious problems appear. Rework and retesting are not the most expensive consequences when that happens. Loss of reputation is.

9 RESULTS

9.1 Performance Metrics

Based on the two types of user recommendations mentioned above, we analyze the performance of all the techniques mentioned above. The resultant jobs recommended to each new user are then checked with the job that the user is originally in as per the test dataset. If the original user job is recommended in the model result, then the model appends 1 for yes else, it appends 0 for no.

This array of 0's and 1's thus received is then checked for accuracy by computing the count of 1's from the total user predictions

Among all the models made with the incorporation of different similarity metrics, the cosine similarity based job recommendation system model outperformed rest of them all. The metrics used to analyse the model performance are: accuracy, precision, recall and F1-score. This is because cosine considers the existence of duplicate terms while computing similarity. Also, computationally, cosine has low complexity and ease over handling sparse data vectors since only non-zero dimensions are considered.

Upon analyzing the result table we observe that the short-comings of some similarity measures upon recommending top 5 and highest-score based job recommendations as even upon achieving high similarity scores is due to the fact that users are seen to have different jobs than the ones recommended by the models, thus resulting in 6–10% error rates.

10 ADVANTAGES

- The model doesn't need any data about other users, since the recommendations are specific to this user.
- This makes it easier to scale to a large number of users.
- The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

DISADVANTAGES

- Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, the model can only be as good as the hand-engineered features.
- The model can only make recommendations based on existing interests of the user.
- In other words, the model has limited ability to expand on the users' existing interests

11 CONCLUSION

In this project, Content-Based Filtering and Collaborative Filtering of recommendations have been compared. Additionally, an aggregation plus recommender system has been devised.

Content-Based Filtering recommends the results based on matching the personal preferences of the user with the given document whereas collaborative filtering recommends based on the preferences of fellow users. On evaluating both of these methods, it was concluded that a hybrid system of both of these overcomes the limitations of both of them and increases the efficiency of ranking. Problems of cold start, sparse database, scalability, and lack of trend recommendation have been eliminated. The proposal is to design a Job Recommender system that prioritizes quality over quantity. While there are websites and job listing portals already recommending jobs to job seekers based on their profiles, this research on aggregate

quality recommendations has been achieved by crawling selectively, overcoming the limitations. A fully functioning user interface was developed to combine everything together to give the user a seamless experience.

12 FUTURE SCOPE

Future works in the case of Personalized Job Recommendation Systems are the utilization of the user-preferred location to get job recommendations based on jobs in organizations established in nearby areas. This can be done by extracting the latitudes and longitudes of the user-preferred location and computing the euclidean distances between the latitudes and longitudes of the organization location.

This filters out other jobs that fall far from the user-preferred location and gives a more accurate job recommender

As part of the future work, we plan to use features of similar candidates and jobs in sequence information. As of now, recommendation using similar candidates and jobs forms part of non-machine learning based recommendations and the initial result seems promising. Finally, it would be interesting to extend our methodology to other recommender systems

13 APPENDIX

Source Code

```

<html><style>
body{
  background-color: #1E135E ;
}
h1{
  font-family: 'Open Sans', sans-serif;
  color: #4BC38A;
}
h2,h3{
  font-family: 'Open Sans', sans-serif;
  color: #FFF;
}
</style>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "15e9c4fc-b3f9-455a-a098-957aceffb9c", // The ID of this
integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "20178e4c-5064-41eb-af43-e9f7bc20beea", // 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>
<center>
<h1>Welcome to Job recommender application</h1>
<h2><u>features of an application</u></h2>
  <h3><li>User registration</li></h3>
  <h3><li>User Login</li></h3>
  <h3><li>Job search</li></h3>
  <h3><li>Job recommendation assistant</li></h3>
  <h3><li>Apply for jobs</li></h3>
  <h3><li>Candidate registration</li></h3>
</center>
</html>
<html>
<head>
  <style>
    #customers {
      font-family: Arial, Helvetica, sans-serif;
      border-collapse: collapse;
      width: 50%;
    }

    #customers td, #customers th {
      border: 1px solid #4BC38A;
      padding: 8px;
      color: #4BC38A
    }

    #customers tr:nth-child(even){background-color: #f2f2f2;}

    #customers tr:hover {background-color: #ddd;}

```



```

#customers th {
    padding-top: 12px;
    padding-bottom: 12px;
    text-align: left;
    background-color: #4BC38A;
    color: white;
}

body{
    font-family: 'Open Sans', sans-serif;
    color: #4BC38A;
    background-color: #1E135E ;
}

</style>
</head>
<body><center><h1>User List</h1>
    <table border="1" id="customers">
        <tr> <th>Names</th><th>Skills</th>  </tr>
        <tr>
            <td>{{ names[0] }}</td>
            <td>{{ skills[0] }}</td>
        </tr>
        <tr>
            <td>{{ names[1] }}</td>
            <td>{{ skills[1] }}</td>
        </tr>
        <tr>
            <td>{{ names[2] }}</td>
            <td>{{ skills[2] }}</td>
        </tr>
    </table>
</body>

```

```
<tr>
  <td>{{ names[3] }}</td>
  <td>{{ skills[3] }}</td>
</tr>
<tr>
  <td>{{ names[4] }}</td>
  <td>{{ skills[4] }}</td>
</tr>
</table>
</center>
</body>
</html>
```

```
<html>
<head><title>JOB SEARCH</title>
<head><style>.c input{
padding: 10px;
width: 30%;
padding-right: 12px 20px;
margin: 8px 0;
bottom: 80px;

display: inline-block;
border: 1px solid #4BC38A;
border-radius: 10px;
box-sizing: border-box;
}
body{
font-weight: bold;
font-family: 'Open Sans', sans-serif;
```

```
        color: #4BC38A;
background-color: #1E135E ;
margin-left: 37%;
margin-top: 5%;
    }
    button {
        font-weight: bold;
        font-family: 'Open Sans', sans-serif;
        background-color: #4BC38A;
        color: #1E135E ;
        padding-right: 12px 20px;
        margin: 8px 0;
        bottom: 80px;
        width: 30%;
        padding: 1%;

        display: inline-block;
        border: 1px solid #4BC38A;
        border-radius: 10px;
        box-sizing: border-box;
    }</style></head>
<body>
    <div class="c">
        <form method="post" action="api">
            <h1>JOB SEARCH</h1><br><br>
            <label>Search for a job.....!</label><br>
            <input type="text" name="search"><br><br>
            <input type="submit">
        </div>
    </body>
</form>
```

```
</html><html>
```

```
<head>
```

```
<title>API</title>
```

```
<style>
```

```
  #customers {
```

```
    font-family: Arial, Helvetica, sans-serif;
```

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

```
    width: 50%;
```

```
  }
```

```
  #customers td, #customers th {
```

```
    border: 1px solid #4BC38A;
```

```
    padding: 8px;
```

```
    color: #4BC38A
```

```
  }
```

```
  #customers tr:nth-child(even){background-color: #f2f2f2;}
```

```
  #customers tr:hover {background-color: #ddd;}
```

```
  #customers th {
```

```
    padding-top: 12px;
```

```
    padding-bottom: 12px;
```

```
    text-align: left;
```

```
    background-color: #4BC38A;
```

```
    color: white;
```

```
  }
```

```
  body{
```

```
    font-family: 'Open Sans', sans-serif;
```

```
        color: #4BC38A;
        background-color: #1E135E ;

    }
</style>

</head>

<body>
    <div class=d>
        <center>
            <h1>Recommended list of jobs for your search...!</h1>
            <table id="customers">
                <tr> <th>Job title</th><th>Company Name</th><th>Location</th><th>Job
Description</th>  </tr>
                <tr>
                    <tr >
                        <td>
                            <h6>JOB Title: {{title1}} </h6>
                        </td>
                        <td>
                            <h6>Comapny Name: {{company1}} </h6>
                        </td>
                        <td>
                            <h6> Location: {{location1}} </h6>
                        </td>
                        <td>
                            <h6> Description: {{description1}} </h6>
                        </td>
                    </tr >
                <tr>
```

```
<td>
    <h6>JOB Title:{{title2}}</h6>
</td>
<td>
    <h6>Comapny Name:{{company2}}</h6>
</td>
<td>
    <h6> Location:{{location2}}</h6>
</td>
<td>
    <h6> Description:{{description2}}</h6>
</td>
</tr>
<tr >
    <td>
        <h6>JOB Title:{{title3}}</h6>
    </td>
    <td>
        <h6>Comapny Name:{{company3}}</h6>
    </td>
    <td>
        <h6> Location:{{location3}}</h6>
    </td>
    <td>
        <h6> Description:{{description3}}</h6>
    </td>
</tr>
<tr >
    <td>
        <h6>JOB Title:{{title4}}</h6>
    </td>
```

```
<td>
    <h6>Comapny Name:{{company4}}</h6>
</td>
<td>
    <h6> Location:{{location4}}</h6>
</td>
<td>
    <h6> Description:{{description4}}</h6>
</td>
</tr>
<tr >
    <td>
        <h6>JOB Title:{{title5}}</h6>
    </td>
    <td>
        <h6>Comapny Name:{{company5}}</h6>
    </td>
    <td>
        <h6> Location:{{location5}}</h6>
    </td>
    <td>
        <h6> Description:{{description5}}</h6>
    </td>
</tr>
</table>
</center>
</div>
</body>

</html>
<html>
```

```
<head>
<style>
#customers {
  font-family: Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 50%;
}

#customers td, #customers th {
  border: 1px solid #4BC38A;
  padding: 8px;
  color: #4BC38A
}

#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}

#customers th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4BC38A;
  color: white;
}
body{
  font-family: 'Open Sans', sans-serif;
  color: #4BC38A;
  background-color: #1E135E ;
}
```



```

</style>
</head>
<body><center><h1>CANDIDATE LIST</h1>
  <table border="1" id="customers">
    <tr> <th>Id</th> <th>Names</th><th>Skills</th> <th>Experience</th>
      <th>Gender</th> <th>Email</th> <th>Phno</th> <th>Address</th>
<th>Salary</th> </tr>
    <tr>
      <td>{{id[0]}}</td>
      <td>{{ names[0] }}</td>
      <td>{{skills[0]}}</td>
      <td>{{experience[0]}}</td>
      <td>{{gender[0]}}</td>
      <td>{{email[0]}}</td>
      <td>{{phno[0]}}</td>
      <td>{{address[0]}}</td>
      <td>{{salary[0]}}</td>
    </tr>
    <tr>
      <td>{{id[1]}}</td>
      <td>{{ names[1] }}</td>
      <td>{{skills[1]}}</td>
      <td>{{experience[1]}}</td>
      <td>{{gender[1]}}</td>
      <td>{{email[1]}}</td>
      <td>{{phno[1]}}</td>

      <td>{{address[1]}}</td>
      <td>{{salary[1]}}</td>
    </tr>
  </table>
</body>
</html>

```

</table>
</center>
</body>
</html>

GitHub Link:

<https://github.com/IBM-EPBL/IBM-Project-13740-1659527949>

Project Demo Link:

<https://www.youtube.com/watch?v=BcyAg1V2f3A>