Project Report Format

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Database Schema (if Applicable)
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

1.INTRODUCTION:

1.1 Project Overview:

The recommender systems are being used in every possible system, for example, clothes recommendation, book recommendation, etc. However, the type of recommendations provided may be different according to the domain of its use. In the case of job recommendation system the case is a little bit different. Here, it will be favorable to provide mostly personalized and profile-based job recommendations. In job recommendation systems, there are varieties of students, having different education levels and skills. Based on students' respective background details, each one of them expects to get only those job recommendations that are highly relevant for that particular student.

1.2 Purpose:

- To make all the users employed according to their suggestions and their skills and job suggestions which are suitable to their opportunities.
- To test the skills and knowledge of the users who needs job.

2.LITERATURE SURVEY:

TITLE	Cost-Effective and Interpretable Job Skill Recommendation with Deep Reinforcement Learning
AUTHORS	Ying Sun, Fuzhen Zhuang, Hengshu Zhu, Qing He, Hui Xiong
YEAR OF PUBLICATION	April 2021
ABSTRACT	Nowadays, as organizations operate in very fast-paced and competitive environments, workforce has to be agile and adaptable to regularly learning new job skills. However, it is nontrivial for talents to know which skills to develop at each working stage. To this end, in this paper, we aim to develop a cost-effective recommendation system based on deep reinforcement learning, which can provide personalized and interpretable job skill recommendation for each talent. Specifically, we first design an environment to estimate the utilities of skill learning by mining the massive job advertisement data, which includes a skill-matching-based salary estimator and a frequent itemset-based learning difficulty estimator. Based on the environment, we design a Skill

	Recommendation Deep Q-Network (SRDQN) with multi-task structure to estimate the long-term skill learning utilities. In particular, SRDQN recommends job skills in a personalized and cost-effective manner; that is, the talents will only learn the recommended necessary skills for achieving their career goals. Finally, extensive experiments on a real-world dataset clearly validate the effectiveness and interpretability of our approach.
METHODOLOGY	Data Mining and Deep reinforcement learning
MERITS	Cost effective
DEMERITS	Must improve the performance for potential application such as curriculum recommendation
OVERCOME DEMERITS	Using CNN for comparable profile to be more potential
LINK	https://doi.org/10.1145/3442381.3449985

TITLE	Prediction of recommendations for employment utilizing machine learning procedures and geo-area based recommender framework
AUTHORS	Binny Parida, Prashanta Kumar Patra, Sthitapragyan Mohanty
YEAR OF PUBLICATION	19 November 2021
ABSTRACT	With increment in the utilization of Internet, the pace of increment of social networks is getting ubiquitous in recent years. This paper focuses on the job portal websites. The research objective of this paper is that the recommender framework takes the abilities from the website and makes suggestion to the candidates with the jobs whose descriptions are coordinating with their profiles the most. This paper additionally presents a short presentation on recommender framework and talks about different categories of this framework. From the start, information is cleaned by expelling the filthy information as extra space and duplicates. Then the job recommendations are made to the target applicants on the basis of their preferences. It utilizes different Machine Learning procedures which results

	show that Random Forest Classifier (RFC) gives the most noteworthy expectation accuracy when contrasted with different procedures. Finally, the optimization technique is utilized to get the most exact outcome. The advantage of recommender framework in career orientation is expressed. Geo-area based recommendation framework is utilized to find the organization's position which can assist the ideal applicants with reaching their destination. This examination shows that the utilization of job recommender system can assist with improving the recommendation of appropriate employment for work searchers
METHODOLOGY	Machine Learning
MERITS	Comparing multiple algorithms.
DEMERITS	Doesn't find for all the close by location organization
OVERCOME DEMERITS	Using Nearest Neighbor algorithm can suggest a close by location organization.
LINK	https://doi.org/10.1016/j.susoc.2021.11.001

TITLE	***A Personalized Brand Proposal Based on User's Satisfaction and Curriculum Supported by an Intelligent Job Recommender System***
AUTHORS	Patricia Rayón Villela , Nelly Rigaud Téllez
YEAR OF PUBLICATION	25 July 2021

ABSTRACT	One of the main challenges' universities are confronted is the personalization of education services to improve quality mechanisms and strategies for supporting and assisting students when entering the workforce. Although many universities try to narrow the gap between academic life and job market, it is a highly challenging task to identify the right job for the right graduate. Market strives to find the most talented people and universities attempt to enrich students' personal brands, but these do not always align. Pitfalls are found in obtaining proper information that harmonize employment offers, course content and graduate's profile. This research places a transversal analysis of job mismatch in Latin American (LATAM) countries, builds a personalized brand based on satisfaction and course content and offers descriptions for an intelligent job recommender system. Proposal considers that providing a targeted job match implies by picking quantitatively relevant technical knowledge and transversal competencies of individual graduates and matching them to knowledge, skills and attitudes of employment offers and course content, in an efficient manner. Competencies from employment offers obtained with text mining are related to those from a current curriculum to help graduates bring about a personal brand for an appropriate job. Contribution of this research is the construction of a framework to construct match patterns that benefits graduates to meet professional success and to achieve personalization and optimization of the universities' offered services that represents an incremental improvement.
METHODOLOGY	Data Mining
MERITS	High Accuracy
DEMERITS	Doesn't have the support for the geo based search
OVERCOME DEMERITS	Using machine learning and deep learning

LINK	10.1007/978-981-16-3941-8_12

TITLE	***Embedding-based Recommender System for Job to Candidate Matching on Scale***
AUTHORS	Jing Zhao, Jingya Wang, Madhav Sigdel, Bopeng Zhang, Phuong Hoang, Mengshu Liu and Mohammed Korayem
YEAR OF PUBLICATION	1 July 2021
ABSTRACT	The online recruitment matching system has been the core technology and service platform in CareerBuilder. One of the major challenges in an online recruitment scenario is to provide good matches between job posts and candidates using a recommender system on the scale. In this paper, we discussed the techniques for applying an embedding-based recommender system for the large scale of job to candidates matching. To learn the comprehensive and effective embedding for job posts and candidates, we have constructed a fused-embedding via different levels of representation learning from raw text, semantic entities and location information. The clusters of fused-embedding of job and candidates are then used to build and train the Faiss index that supports runtime approximate nearest neighbor search for candidate retrieval. After the first stage of candidate retrieval, a second stage reranking model that utilizes other contextual information was used to generate the final matching result. Both offline and online evaluation results indicate a significant improvement of our proposed two-staged embedding based system in terms of click-through rate (CTR), quality and normalized discounted accumulated gain (nDCG), compared to

	those obtained from our baseline system. We further described the deployment of the system that supports the million-scale job and candidate matching process at CareerBuilder. The overall improvement of our job to candidate matching system has demonstrated its feasibility and scalability at a major online recruitment site.
METHODOLOGY	Deep learning
MERITS	High Accuracy
DEMERITS	Didn't compared with many algorithms.
OVERCOME DEMERITS	Compare more models with the same data.
LINK	https://doi.org/10.48550/arXiv.2107.00221

TITLE	***Job Recommendation based on Job Profile Clustering and Job Seeker behaviour.***
AUTHORS	D. Mhamdi*, R. Moulouki, M. Y. El Ghoumari, M. Azzouazi, L. Moussaid
YEAR OF PUBLICATION	August 2020
ABSTRACT	This article presents a recommender system that aims to help job seekers to find suitable jobs. First, job offers are collected from job search websites then they are prepared to extract meaningful attributes such as job titles and technical skills. Job offers with

	common features are grouped into clusters. As job seeker like one job belonging to a cluster, he will probably find other jobs in that cluster that he will like as well. A list of top n recommendations is suggested after matching data from job clusters and job seeker behavior, which consists on user interactions such as applications, likes and rating
METHODOLOGY	Clustering and Artificial Intelligence
MERITS	Good Accuracy
DEMERITS	Doesn't find perfect job with the required user skill
OVERCOME DEMERITS	using Word2vec method and k-means clustering algorithms used to capture and represent the context of job profiles
LINK	https://doi.org/10.1016/j.procs.2020.07.102

2 LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. Confusing Application Process. Each job advertisement will have its own guidelines for filling out the application.
2. Staying Up to Date.
3. Having a Limited Professional Network.
4. Not Having the Right Degree.
5. No Feedback.

2.2 REFERENCES

1. [PDF] Recruitment and Job Search Application (researchgate.net)

2. (99+) Mobile Application System for Online Job seeker | vishakha Nagrale - Academia.edu
3. https://core.ac.uk/download/pdf/77979433.pdf

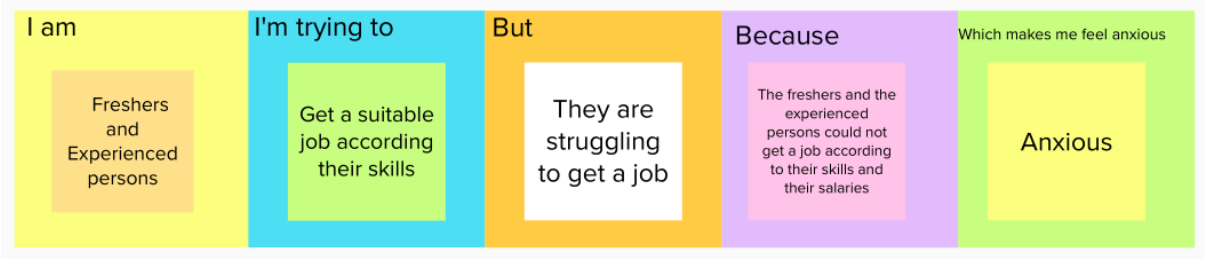
2.3 PROBLEM STATEMENT:

Date	19 September 2022
Team ID	PNT2022TMID27298
Project Name	Project – SKILL AND JOB RECOMMENDER
Maximum Marks	2 Marks

Customer Problem Statement Template:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Freshers	Get a suitable job according to their skills	They are suffering to get a suitable job.	The freshers could not get a job according to their relevant skills.	Anxious

PS-2	Experienced persons	Get a job based on their skills and get a better salary	They are not getting better salaries according to their skills.	Experienced people could not get better salaries according to their skills.	Anxious

3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map canvas:-

3.2 Ideation & Brain Storming:-

Date	19 September 2022
Team ID	PNT2022TMID27298
Project Name	Skill and Job Recommender
Maximum Marks	4 Marks

Brainstorm & Idea Prioritization Template:

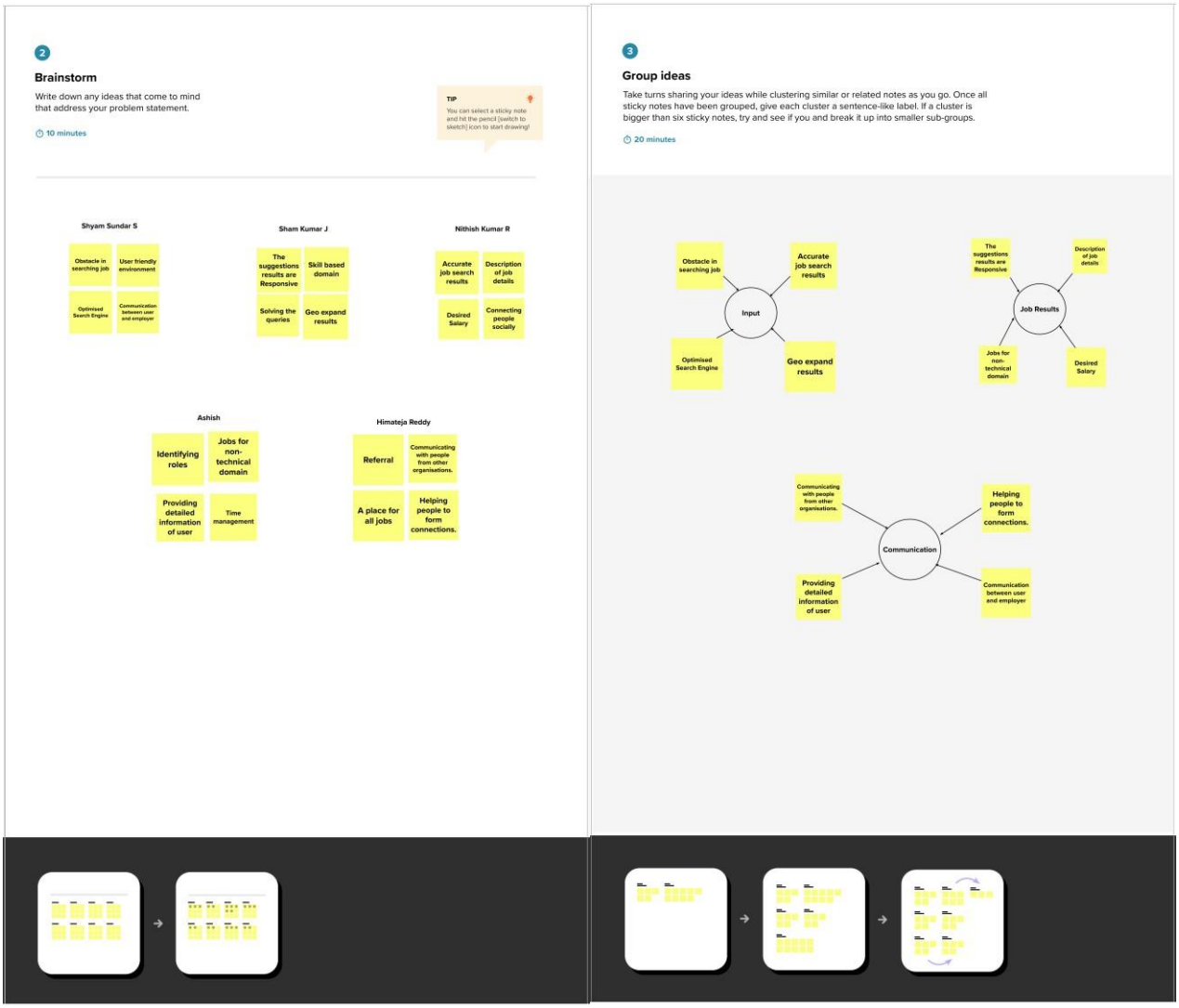
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

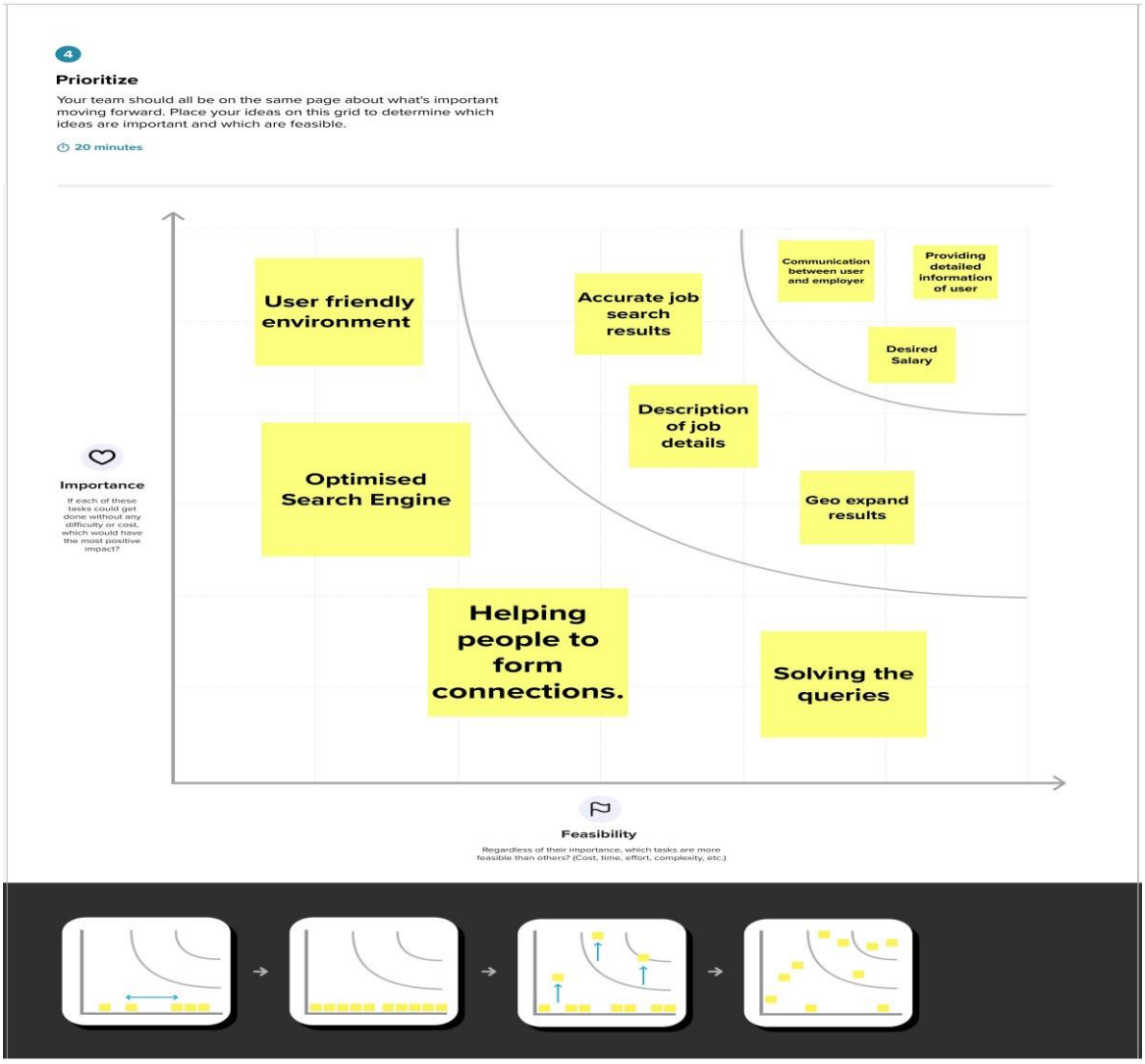
Reference: https://www.mural.co/templates/empathy-map-canvas

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 Proposed Solution :-

Date	September 2022
Team ID	PNT2022TMID27298
Project Name	Skill and Job Recommender
Maximum Marks	2 Marks

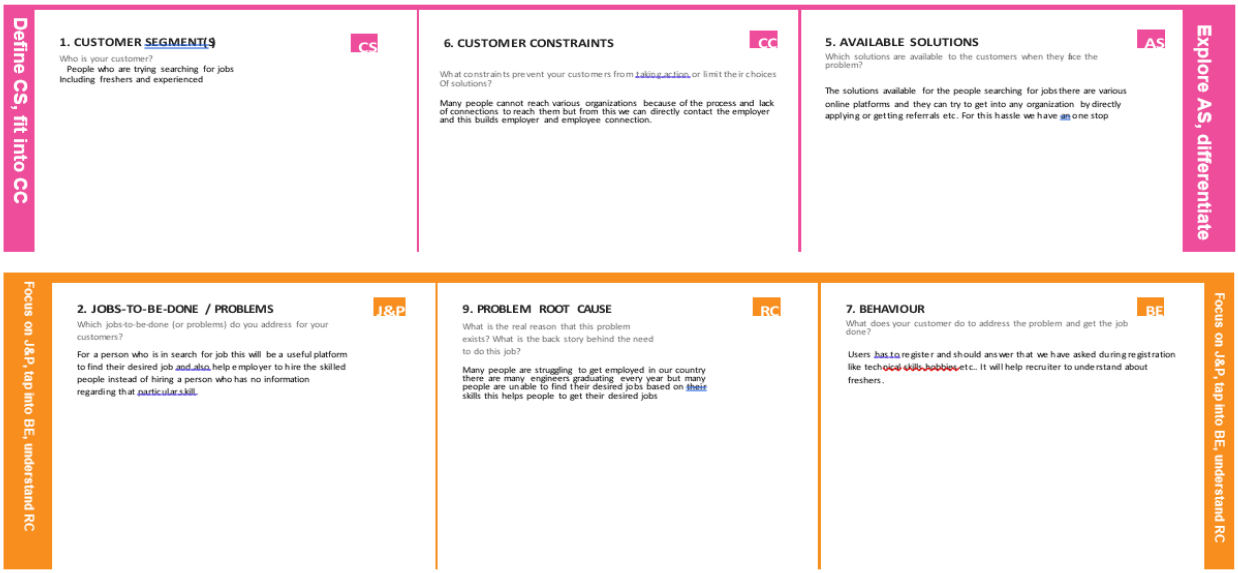
Proposed Solution Template:

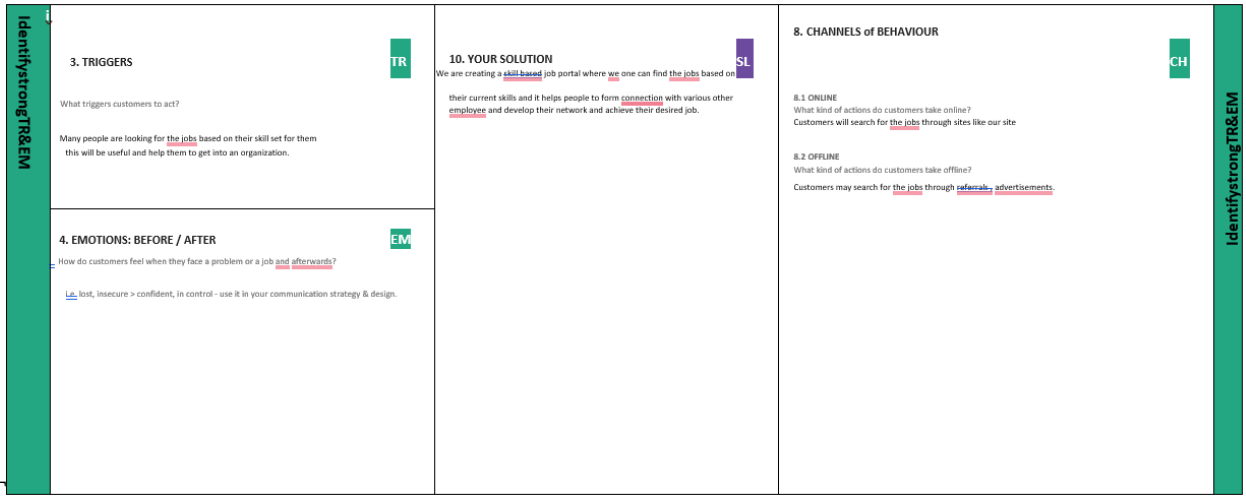
Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The web application doesn't have all the relevant jobs that user seeks.
2.	Idea / Solution description	Based on the person-job fit premise, we propose a framework for job recommendation based on professional skills of job seekers. We automatically extracted the skills from the job seeker profiles using a variety of text processing techniques
3.	Novelty / Uniqueness	The skills and the job recommendation system is basic way of job recommendations by matching with the manually entered skills in the existing research projects. In this project the skills and the jobs are identified using the data taken from the student and the jobs are recommended based on the eligibility criteria with the specified skills.
4.	Social Impact / Customer Satisfaction	All skilled employees and fresher's are got employed using the recommender systems. Suitable jobs are suggested according the skills. The end users can choose their own interested jobs. For the experienced persons it's an easy way to get jobs instead of searching outside.
5.	Business Model (Revenue Model)	For 6 months, The user can access each and every feature in the web application for free and after that they have to subscribe for premium in order to continue the services.

6.	Scalability of the Solution	The recommender system will be adaptable based on the persons who are applying for the jobs. The infrastructure of the application should be maintained without the bugs. The ideas of the recommendation systems will be upgraded to the next level. The performance of the application will be effective and suitable to the hardware systems.

3.4 Problem Solution Fit:-



REQUIREMENT 

4.REQUIREMENT ANALYSIS(FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS):-

Date	03 October 2022
Team ID	PNT2022TMID27298
Project Name	Skill and Job Recommender
Maximum Marks	4 Marks

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 Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Organisation Details	Details about organisation Details about vacancy of job
FR-4	Searching Job	Job details in clear manner. Variety of domains
FR-5	Optimised Details	Details of organisation or job in clear manner Optimised results for searching job.

4.2 Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It's an invaluable tool for job searches and recruitment.
NFR-2	**Security**	The security will in form of passwords, otp or the question is asked.
NFR-3	**Reliability**	It can be reliable because the user will be posting about their education and also certifications
NFR-4	**Performance**	Performance can be calculated based how many users are benefitted through site and also how user feel about the site.
NFR-5	**Availability**	The services like searching for job and applying for job always available
NFR-6	**Scalability**	This recommender system provides job to all people who are in need of job

5 PROJECT DESIGN :-

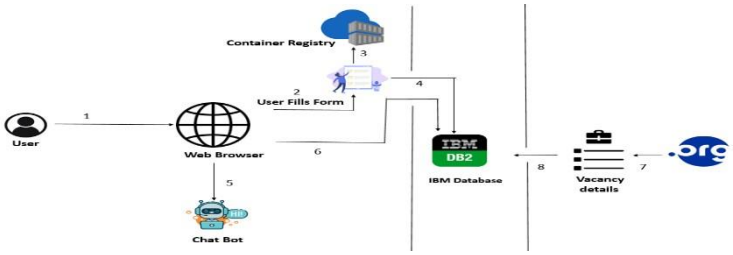
5.1 DATA FLOW DIAGRAMS:-

Date	20 October 2022
Team ID	PNT2022TMID27298
Project Name	Skill and Job Recommender
Maximum Marks	4 Marks

Data Flow Diagrams:-

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

Flow Diagram :-



1)User will login into website and search for job.

2)User fills form for applying job.

3)The list of users will be stored in container registry.

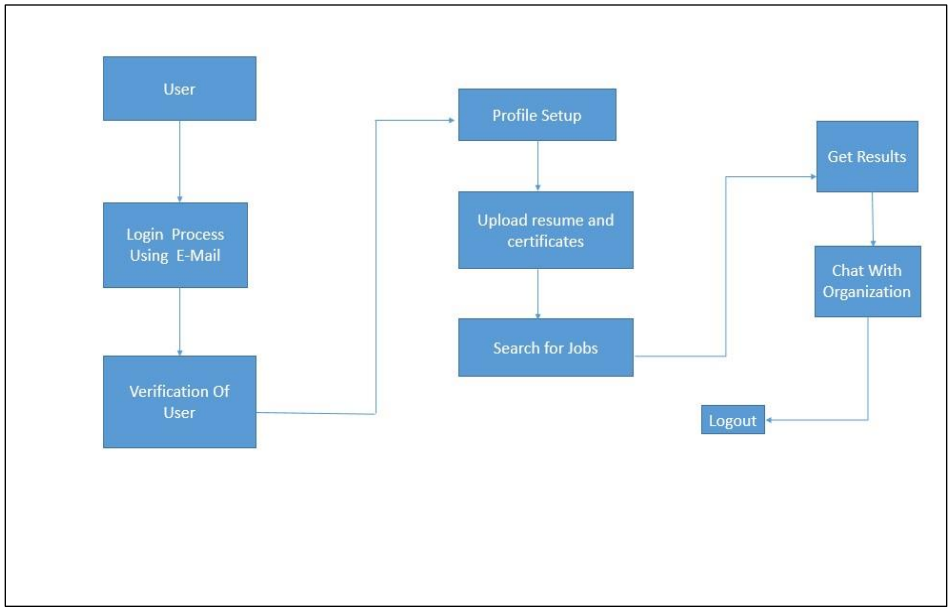
4)Data will be stored in IBM DB.

5)Chat Bot will display the results of job according to user skills.

6)For job recommendation it will fetch details from database.

7)Organization will provide vaccancy details.

8) Vacancy details stored in IBM Database



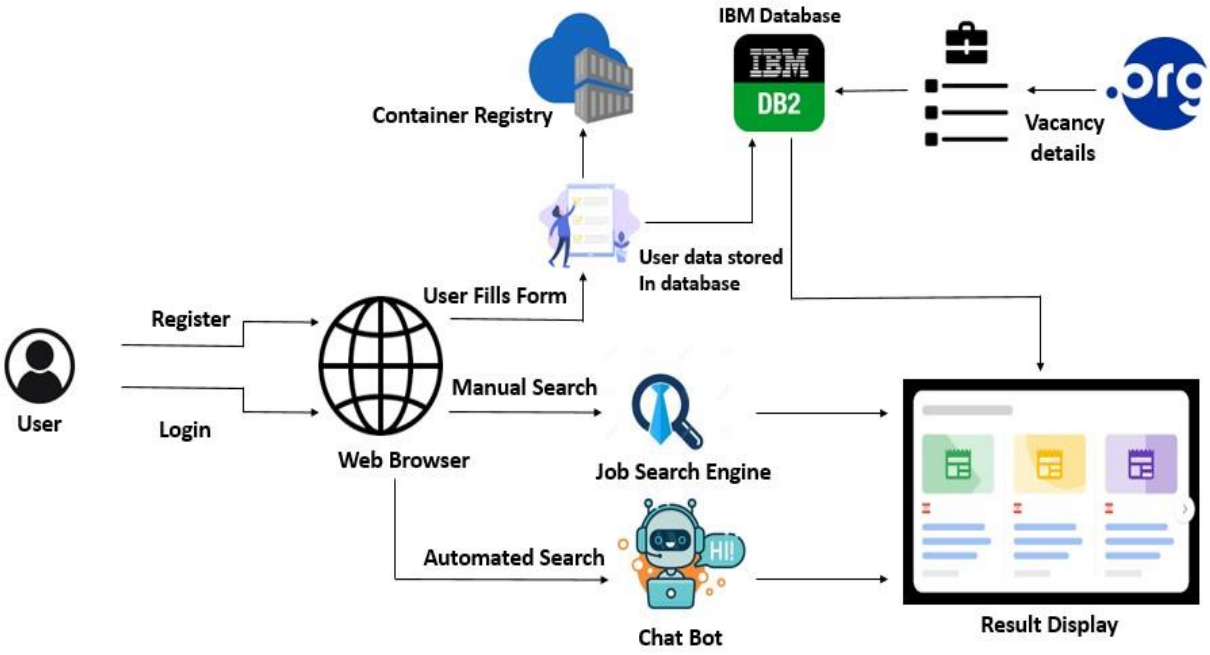
5.2 SOLUTION & TECHNICAL ARCHITECTURE :-

Date	19 September 2022
Team ID	PNT2022TMID27298
Project Name	Skill and Job Recommender
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.



5.3 USER STORIES:-

User Stories :-

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

pe	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Registration	USN-1	As a user, I can register f by entering my email, password, and confirming my password, skills required for job, hobbies, languages known, experiences.	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	s a user, I can register for the pplication through LinkedIn	I can register & access the dashboard with LinkedIn	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail.	Medium	Sprint-1
	Applying Job	USN-5	As a user, I can apply for the job using the input form which is registered during the registration process.	I can apply for job by matching with the form I registered and the job applying criteria given by organization	High	Sprint-1

	Changing Domain	USN-6	As a user, I can also change my domain which is different my course.	I can apply for job in different domains by filtering the domains option	High	Sprint-1
ce	Applying Job	USN-1	As an experienced person, I can apply for job using the experience which I have been included in the form (resume).	I can apply for the higher positions by comparing thee experience I have with company job criteria	High	Sprint-1
	Selecting based on roles	USN-2	As an experienced person, I can apply for job using the experience which I have been included in the form (resume).	I can change the roles using the job filter drop down menu.	Medium	Sprint-2
	Selecting based on salary package	USN-3	As an experienced person, I can expect the desired salary package.	I can expect the desired salary package based on the experience or previous job salary package	High	Sprint-1

e	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
	Issuing offer letter	USN-1	As a customer executive, I can rectify the user issue if the offer letter is not issued or any technical issues.	Offer letter is not received on the correct date	High	Sprint-1
	Acknowledgement mail	USN-2	As a customer executive, if the customer issue has been rectified acknowledgement should be sent	Acknowledgement can sent using mail or message	High	Sprint-2
ato	Technical Issues	USN-1	As a admin, I have to solve any technical glitch happened like if the user data has been leaked or any malware has been there in server.	Issues can be rectified by scanning the entire system	High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING :-

	Functional Requirement (Epic)	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.

		USN-2	As a user, I will receive a confirmation email once I have registered for the application.	1	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
		USN-3	As a user, I can register for the application through Facebook.	2	Low	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
		USN-4	As a user, I can register for the application through Gmail.	2	Medium	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	Login	USN-5	As a user, I can log into the application by entering my email & password.	1	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	Dashboard	USN-7	As a user, I can access the website in a second.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	Dashboard	USN-8	As a user, If I Log in correctly, I can view my dashboard and I can navigate to any pages which are already listed there.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar
						R, Ashish M, Pelleti Hima Teja Reddy.
	User Profile	USN-9	As a user, I can view and update my details.	2	Medium	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.

	Functional Requirement (Epic)	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
	Database	USN-10	As a user, I can store my details and data in IBM Database.	2	Medium	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	Cloud Storage	USN-11	As a user, I can upload my photo, resume and much more in the website.	1	Medium	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	Chatbot	USN-12	As a user, I can ask the Chatbot about the latest job openings, which will help me and show the recent job openings based on my profile.		High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.

	Identity-Aware	USN-13	As a User, I can access my account by entering the correct login credentials and my user credentials are only displayed to me.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	SendGrid service	USN-14	As a user, I can get a notification or mail about a job opening with the help of the SendGrid service.	1	Medium	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	Learning Resource	USN-15	As a user, I can learn the course and I will attain the skills which will be useful for developing my technical skills.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	Docker	USN-16	As a user, I can access the website in any device.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	Kubernetes	USN-17	As a user, I can access the website in any device.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.

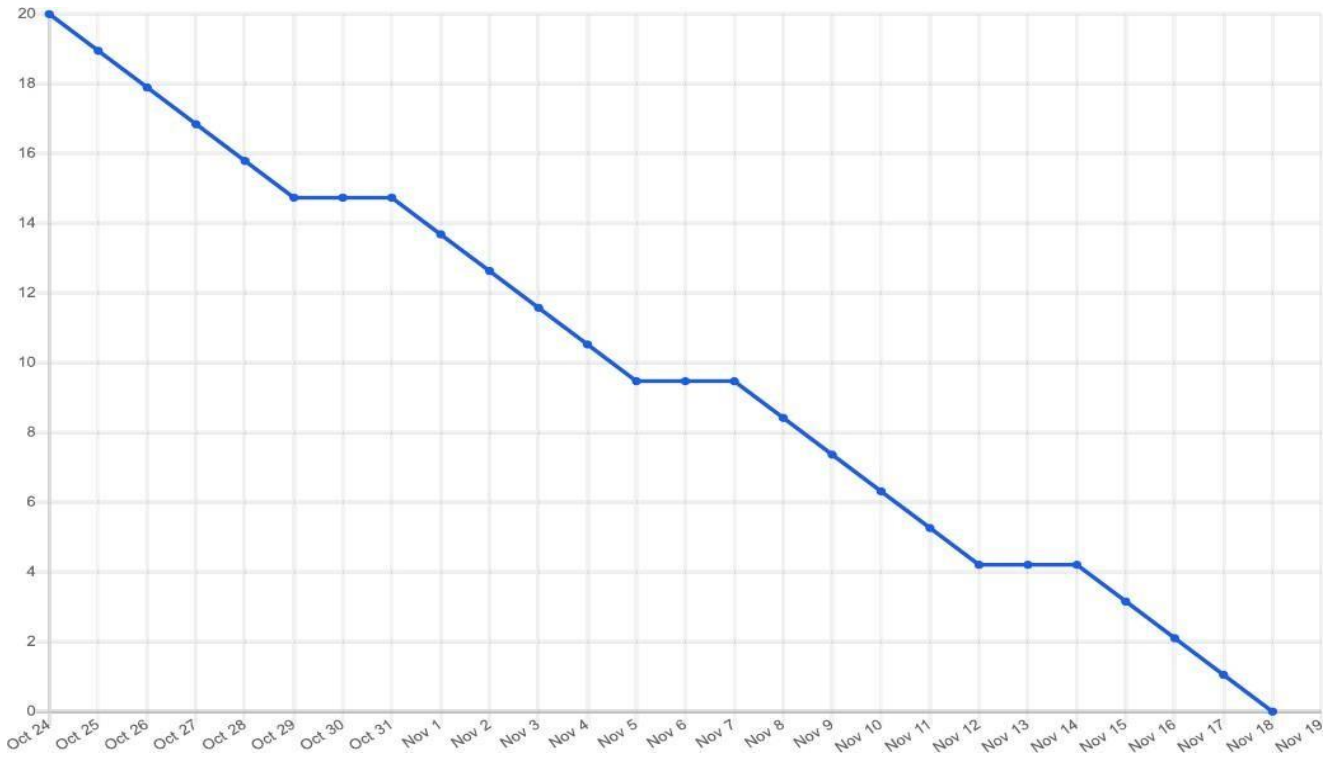
Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
20	6 Days	31 Oct 2022	05 Nov 2022	19	05 Nov 2022
20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

t-3	Deployment in cloud	USN-18	As a user, I can access the website in any device.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
t-3	Technical support	USN-19	As a user, I can get a customer care support from the website which will solve my queries.	1	Medium	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
t-4	Unit Testing	USN-15	As a user, I can access the website without any interruption.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.

	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members

	Integration testing	USN-16	As a user, I can access the website without any interruption.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.
	System testing	USN-17	As a user, I can access the website without any interruption.	2	High	Shyam Sundar S, Sham Kumar J, Nithish Kumar R, Ashish M, Pelleti Hima Teja Reddy.

Project Tracker, Velocity & Burndown Chart: (4 Marks) **Burndown Chart:**



CODING AND SOLUTION

1 FEATURE 1

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Skill Login</title>
    <style>
        *{
```

```
    margin: 0;
    padding: 0;

    box-sizing: border-box;
}

body{
      background-color: cornflowerblue;
    width: 100%;
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
}
.container{
    width: 50%;
    height: 60vh;
    padding: 20px;
    border-radius: 20px;
    box-shadow:  0px 5px 25px rgba(0,0,0,0.2);
    display: flex;
    justify-content: space-evenly;
    align-items: center;
    flex-direction: column;
   background-color: rgb(211,211,211);
}

.container form{
    width: 100%;
    height: 100%;
    display: flex;
    justify-content: space-evenly;
    align-items: center;
    flex-direction: column;
}

.inputBox{
    width: 100%;
    margin: 10px 0;
    border-radius: 5px;
    overflow: hidden;
    height:50px;

}

.inputBox input[type="text"], .inputBox input[type="email"]{
    width: 100%;
    height: 50px;
    border-radius: 5px;
    border: none;
    outline: none;
    overflow: hidden;
    border: 1px solid rgba(0,0,0,0.2);
    padding: 0px 10px;
    font-size: 16px;
    color: #444;
}

.inputBox textarea{
    width: 100%;
```

```
    height: 120px;
    border-radius: 5px;
    border: none;
    outline: none;
    overflow: hidden;
    border: 1px solid rgba(0,0,0,0.2);
    padding: 0px 10px;
    font-size: 16px;
    color: #444;
}

.inputBox button{
    width: 100%;
    padding: 10px 20px;
    border: none;
    outline: none;
    background: rgb(0, 119, 255);
    color: #FFF;
    font-size: 20px;
    font-weight: bold;
    cursor: pointer;

}

.inputBox button:hover{
    background: rgb(0, 17, 255);
    transition: all 0.3s ease;
}

::placeholder{
    font-size: 16px;
}

.alert{
    width: 100%;
    background: rgb(0, 255, 106);
    padding: 10px 20px;
    border-radius: 5px;
    text-align: center;
    font-size: 18px;
    font-weight: 900;
    display: none;
}
h1{
color:cornflowerblue;
}
    </style>
</head>
<body>
    <div class="container">

      <b><h1>IBM</h1></b>
        <form action="/login" method="POST">

            <div class="inputBox">
                <input type="text" id="emailid" name="fn" placeholder="Enter Your Name" />
            </div>
            <div class="inputBox">
                <input  class="inputBox" type="password" id="name" name="password"
placeholder="Enter Password...." />
```

```
            </div>



            <div class="inputBox">
                <button type="submit">Submit</button>
            </div>
            <center><h4>Return to <a href="homepage.html">Homepage</a></h4></center>
        </form>
    </div>

   <script>
  window.watsonAssistantChatOptions = {
    integrationID: "a2445f85-b7a3-4c58-980c-72a414ea407b", // The ID of this integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "1c2b79b4-a6d1-4c4d-bf2f-852aa415e243", // 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>
</body>

</html>
```

1. Above code is for the login feature
2. It is to ensure that the correct user user login into the portal.

```
3. from flask import Flask, render_template, request
   import ibm_db

   app = Flask(__name__)

   conn = ibm_db.connect(
       "DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-
   ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLServer
   Certificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=dtk93338;PWD=honkzW80jUXH3UJN;",
       "", "")
   num=[]
   num1=[]
   num2=[]

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

       return render_template('homepage.html')

   @app.route('/login.html')
   def login1():
       return render_template('login.html')
```

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

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

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

@app.route('/login',methods=["GET","POST"])
def login():
     i = int(0)
     if request.method=="POST":
        fn= request.form.get('fn')
        sql = "select * from userdetails"
        stmt = ibm_db.exec_immediate(conn, sql)
        while ibm_db.fetch_row(stmt) != False:
           em=ibm_db.result(stmt,0)
           num1.append(em)
           i=i+1

        for y in range(i):
           ele2=str(fn)
           if ele2==num1[y]:
              return render_template("apply.html")




@app.route('/register', methods=["GET", "POST"])
def insert():
    i=int(0)
    if request.method == "POST":
        fn = request.form.get('firstname')
        mn = request.form.get('middlename')
        ln = request.form.get('lastname')
        course = request.form.get('course')
        gender = request.form.get('gender')
        #skill = request.form.get('skillset')
        email = request.form.get('email')
        pw = request.form.get('password')
        sql = "select * from userdetails"
        stmt = ibm_db.exec_immediate(conn, sql)
        while ibm_db.fetch_row(stmt) != False:
           em=ibm_db.result(stmt,0)
           num.append(em)
           i=i+1
        for y in range(i):
           ele2=str(fn)
           if ele2==num[y]:
              print("already registered mail")
              return render_template("alertwrong.html")


        insert_sql = "Insert INTO userdetails Values(?,?,?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
```

```
            ibm_db.bind_param(prep_stmt, 1, fn)
            ibm_db.bind_param(prep_stmt, 2, mn)
            ibm_db.bind_param(prep_stmt, 3, ln)
            ibm_db.bind_param(prep_stmt, 4, course)
            ibm_db.bind_param(prep_stmt, 5,  gender)
            ibm_db.bind_param(prep_stmt, 6, email)
            ibm_db.bind_param(prep_stmt, 7, pw)
            ibm_db.execute(prep_stmt)
            print("inserted")
            return render_template('register.html')

    @app.route('/apply',methods=["GET","POST"])
    def applyforjob():
        fn = request.form.get('fn')
        ln = request.form.get('ln')
        citizenship = request.form.get('Citizenship')
        city = request.form.get('city')
        twelve = request.form.get('twelth')
        tenth = request.form.get('tenth')
        insert_sql = "insert INTO newapply values(?,?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, fn)
        ibm_db.bind_param(prep_stmt, 2, ln)
        ibm_db.bind_param(prep_stmt, 3, citizenship)
        ibm_db.bind_param(prep_stmt, 4, city)
        ibm_db.bind_param(prep_stmt, 5, twelve)
        ibm_db.bind_param(prep_stmt, 6, tenth)
        ibm_db.execute(prep_stmt)
        return render_template("apply.html")

    if __name__ == '__main__':
        app.run(host='0.0.0.0')
```

The above code is implemented using python language

1. The above code contains all the features like login to ensure privacy
2. It also have the code for registration
3. It also connects to database which ibm database(db2)
4. It has the database of companies which contains job offers ,salary package.

8 TESTING

8.1 TEST CASE

TEST MODULE	TEST CASE	EXPECTED RESULT	TEST RESULT
ADMIN	Provide valid login credentials	User successfully logged in and directed to the admin dashboard page	PASS
ADMIN	Enters invalid login credentials	Displays Error message	PASS
ADMIN	Upon successful login, click on the 'List of Employers' tab.	Displays the details of list of active employers registered with the application	PASS
ADMIN	Click on 'Active/Deactivate' tab under status of the employer	The status of the employer will be changed to active/deactivate.	PASS
EMPLOYER	Provide details for registration	Employer successfully registered with the application	PASS
EMPLOYER	Upon successful login, click on 'Post New Job' tab	Employer posts jobs with the required details	PASS
EMPLOYER	Employer trying to post job with insufficient details	Prompts to fill in all the necessary details of the job	PASS
EMPLOYER	Employer clicks on the 'List Posted Jobs' tab	All the jobs posted by the employer will be displayed.	PASS

EMPLOYER	Employer clicks on ‘Active/deactivate’ under Status	The status of the job posting will changed to active/deactivated.	PASS
EMPLOYER	Employer clicks on the ‘view’ tab under candidates column	The list of the details of applicants for a particular job posting are displayed.	PASS

8.2 USER ACCEPTANCE TESTING

JOB BSEEKER	Provide details for registration	Jobseeker successfully registered with the application	PASS
JOBSEEKER	Enters invalid login credentials	Error message displayed	PASS
JOBSEEKER	Upon successful login, click on ‘My Profile’ tab	List details of jobseeker	PASS
JOBSEEKER	Upon successful login, click on ‘Search Jobs’ tab	Details of the active job postings are displayed.	PASS
JOBSEEKER	Upon successful login, click on ‘Applied Jobs’ tab	Details of the jobs that are applied by the jobseeker are displayed	PASS
JOBSEEKER	Click on ‘Add Review’ tab	Displays a form to fill in the review details of the company	PASS
JOBSEEKER	Logout	Redirects to the Home page of the application	

9 RESULTS

9.1 PERFORMANCE METRICS

Performance testing is performed to determine how well the system can perform in terms of responsiveness under all kinds of load. The web application is tested to see if it can sustain huge amount of requests providing higher throughput under different loads. I have simulated multiple hits on various pages of the application to evaluate the overall performance.

Operating System	Windows 10 (64 bit)
RAM	8 GB
Processor	Intel core i7
Processor Speed	3.40 GHz

10. ADVANTAGES AND DISADVANTAGES:-

ADVANTAGES:-

- Career growth
- Nervousness will be reduced
- Increase in speed of work
- Hardwork
- Confidence

DISADVANTAGES:-

- Not doing the task within deadline
- Not having the necessary skill
- Making mistakes in task
- Stage fear

11. CONCLUSION

- 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.

- Along with this, testing and collecting more user data for better performance of the collaborative filtering module is required. Lastly, improving the cleansing process of the job description and using natural language processing are required. While using collaborative filtering, this work can be improved by giving different weights to different users based on their LinkedIn skills.

12.FUTURE SCOPE:-

- There is a lot of scope of enhancement to the existing recommendation system. An approach naturally solves the candidate and job cold-start problem in the absence of interaction data. The existing system has a job recommendation based on their skills and working field mentioned as a requirement.
- As part of the future work, we can make a good UI for this system with some features like the portal can send email notifications to candidates about certain job availabilities, there can be a feedback or review section for the application.

13.APPENDIX:-

13.1 SOURCE CODE:-

Login.html

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Skill Login</title>
    <style>
        *{
    margin: 0;
```

```
    padding: 0;

    box-sizing: border-box;
}

body{
      background-color: cornflowerblue;
    width: 100%;
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
}
.container{
    width: 50%;
    height: 60vh;
    padding: 20px;
    border-radius: 20px;
    box-shadow:  0px 5px 25px rgba(0,0,0,0.2);
    display: flex;
    justify-content: space-evenly;
    align-items: center;
    flex-direction: column;
   background-color: rgb(211,211,211);
}

.container form{
    width: 100%;
    height: 100%;
    display: flex;
    justify-content: space-evenly;
    align-items: center;
    flex-direction: column;
}

.inputBox{
    width: 100%;
    margin: 10px 0;
    border-radius: 5px;
    overflow: hidden;
    height:50px;

}

.inputBox input[type="text"], .inputBox input[type="email"]{
    width: 100%;
    height: 50px;
    border-radius: 5px;
    border: none;
    outline: none;
    overflow: hidden;
    border: 1px solid rgba(0,0,0,0.2);
    padding: 0px 10px;
    font-size: 16px;
    color: #444;
}
```

```
.inputBox textarea{
    width: 100%;
    height: 120px;
    border-radius: 5px;
    border: none;
    outline: none;
    overflow: hidden;
    border: 1px solid rgba(0,0,0,0.2);
    padding: 0px 10px;
    font-size: 16px;
    color: #444;
}

.inputBox button{
    width: 100%;
    padding: 10px 20px;
    border: none;
    outline: none;
    background: rgb(0, 119, 255);
    color: #FFF;
    font-size: 20px;
    font-weight: bold;
    cursor: pointer;

}

.inputBox button:hover{
    background: rgb(0, 17, 255);
    transition: all 0.3s ease;
}

::placeholder{
    font-size: 16px;
}

.alert{
    width: 100%;
    background: rgb(0, 255, 106);
    padding: 10px 20px;
    border-radius: 5px;
    text-align: center;
    font-size: 18px;
    font-weight: 900;
    display: none;
}
h1{
color:cornflowerblue;
}
    </style>
</head>
<body>
    <div class="container">

      <b><h1>IBM</h1></b>
        <form action="/login" method="POST">

            <div class="inputBox">
```

```
                <input type="text" id="emailid" name="fn" placeholder="Enter
Your Name" />
            </div>
            <div class="inputBox">
                <input  class="inputBox" type="password" id="name"
name="password" placeholder="Enter Password...." />
            </div>



            <div class="inputBox">
                <button type="submit">Submit</button>
            </div>
            <center><h4>Return to <a
href="homepage.html">Homepage</a></h4></center>
        </form>
    </div>

   <script>
  window.watsonAssistantChatOptions = {
    integrationID: "a2445f85-b7a3-4c58-980c-72a414ea407b", // The ID of this
integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "1c2b79b4-a6d1-4c4d-bf2f-852aa415e243", // 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>
</body>

</html>
```

Register.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Registration</title>
<style>
    @import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=sw
ap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
```

```
}
body{
  background:rgba(0,174,239,0.8);
  display:flex;

}
.wrapper{
  width: 500px;
  padding: 2rem 0 1rem 0;
  margin: 50px auto;
  background: #fff;
  border-radius: 10px;
  text-align: center;
  box-shadow: 0 20px 35px rgba(0, 0, 0, 0.1);
}
h1{
  font-size: 2rem;
  background: linear-gradient(to right,cornflowerblue,skyblue,blue);
            -webkit-background-clip: text;
            color: transparent;

}
p{
  margin-bottom: 1.7rem;
}
.inputs{
  width: 85%;
  outline: none;
  border: none;
  background: #dfe9f5;
  padding: 12px 14px;
  margin-bottom: 10px;
  border-radius: 10px;
}

button{
  font-size: 1.1rem;
  margin-top: 1rem;
  padding: 8px 0;
  border-radius: 5px;
  outline: none;
  border: none;
  width: 85%;
  background: cornflowerblue;
  color: #fff;
  cursor: pointer;
}
button:hover{
  background: rgba(100,149,237,0.767);
}
   .radio{
      align-items: center;
   }
   textarea{
      resize: none;
   }
   .email{
```

```
   padding-left:45px;
   }
</style>
</head>
<body>
  <div class="wrapper">
    <h1>Sign up</h1> <br>
    <form action="/register" method="POST">

<label> Firstname </label>
<input class="inputs" type="text" name="firstname" size="15"/> <br> <br>
<label> Middlename: </label>
<input class="inputs" type="text" name="middlename" size="15"/> <br> <br>
<label> Lastname: </label>
<input class="inputs" type="text" name="lastname" size="15"/> <br> <br>
      <label>
Course :
</label>
<select>
<option name="course" value="Course">Course</option>
<option name="course" value="BCA">BCA</option>
<option name="course" value="BBA">BBA</option>
<option name="course" value="B.Tech">B.Tech</option>
<option name="course" value="MBA">MBA</option>
<option name="course" value="MCA">MCA</option>
<option name="course" value="M.Tech">M.Tech</option>
</select>
 <br>
<br>
<label>
Gender :
</label>
<select>
<option name="gender" value="Course">Male</option>
<option name="gender" value="BCA">Female</option>
<option name="gender" value="BBA">Others</option>
</select>
<br>
<br>
      <label>Skillset</label><br>
      <textarea cols="50" rows="5" name="skillset"></textarea><br>

      <label> Password </label>
      <input type="password" name="pn"/><br><br>
      <div class="email">
      <label>Email</label>
      <input type="email" name="email">
      </div>
     <button>Submit</button>
    </form>
      </div>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "a2445f85-b7a3-4c58-980c-72a414ea407b", // The ID of this
integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "1c2b79b4-a6d1-4c4d-bf2f-852aa415e243", // 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>

</body>
</html>
```

Company detai;ls.html

```
<html>
   <head><title>List of Jobs</title>
   <style>
      body {
  width: 100%;
  font-family: sans-serif;
  background-color: antiquewhite;
}

.projects-section {
  margin: 3em;
}

h2 {
  font-size: 2.4em;
  margin-bottom: 32px;
  font-family: sans-serif;
  color: maroon;
}

.card-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 30px;
}

.card {
  display: flex;
  flex-direction: column;
  gap: 12px;
  max-width: 300px;
  overlow: hidden;
}

.card:focus-visible, .card:hover {
  scale: 1.02;
  transition: 400ms ease-in-out;
  outline-offset: 6px;
```

```
  border-radius: 6px;
}

img {
  aspect-ratio: 1/1.2;
  border-radius: .4em;
}

h3 {
  font-size: 1.4rem;
  font-weight: bold;
  font-family: cursive;
}

p {
  font-size: 1rem;
  color: #888;
  line-height: 145%;
}

.link {
  display: inline-flex;
  width: fit-content;
  align-items: center;
}

.link::after {
  display:block;
  content: '';
  border-bottom: solid 3px #019fb6;
  transform: scaleX(0);
  transition: transform 250ms ease-in-out;
}

a {
  color: darkred;
  text-decoration: none;
  font-weight: 500;
}

.arrow {
  padding-left: 4px;
  color: darkred;
}

@media screen and (min-width: 320px) {
  html {
    font-size: calc(16px + 6 * ((100vw - 320px) / 680));
  }
}
@media screen and (min-width: 1000px) {
  html {
    font-size: 18px;
  }
}
      </style>
   </head>
```

```
   <body>
<div class="projects-section">
  <h2>List of Jobs</h2>
  <div class="card-container">

    <div tabindex=0 class="card">
      <img
```
src="
RAVDxAVDw8VDxUVFRAPFRUWFhUXFRgYHSggGBolHRUVITEhJSktLi4uFx8zODMsNygtLisBCgoKDg
0OGhAQFy0dHR8tLS0tLS0tLSstLS0tLS0tLS0tLS0tLS0tLS0tLS0tListLS0tLS0tLS0tLS0
tLf/AABEIALcBEwMBIgACEQEDEQH/xAAcAAABBQEBAQAAAAAAAAAAAAACAAEDBQYEBwj/xABCEAAC
AQIDBAgDBQYFAwUAAAABAgADEQQSIQUGMVETMkFhcYGRoSJSsTNCYsHRFCNygpLCB1NjouEWQ7IVJ
HPw8f/EABoBAAIDAQEAAAAAAAAAAAAAAAABAgQFAwb/xAA3EQACAQIDBQQIBQUBAAAAAAAAAQIDEQ
QhMQUSUZGhYXGx0SIyM0FCgeHwBhRiwdIVNIKSokP/2gAMAwEAAhEDEQA/AOeCYdorTdRlkcYyS0E
iMADAMlIgEQAEwTJCIxEYiMwTJCIJEYiIxoZEa0YAmNCjWgIGCYZEG0QgDGMkIgkRjZGYJEkIgWjs
FwDBIkhEEiIZHGhkRiIAARBMkIjWgBGY9odogpPCIZHBMmWkxNgCT6zpp7NY9Yhf0tZVlTBDUx+0r18TRoe1m
o97z+S1fyRYo4atX9nBy8Oei+bRXmMqkmwBJ5WvLpNn0xxBbPD0Eni+bRXmqkmwBJ5WvLpNn0xxBbPD0Eni
K8+iNajsGrL2s1HuzfkubKens6oeICjv4+gnTT2bTHWBPfwHtO7ibDUngBtqT4Cd2H2JianCnlQdF
9Q2pqzH8Kk28bcJtMPsDDxxnbN/t0aD2mmxAXPf+ks8Ns6vvV61ovaraPfkV6RdLNnDvOenbk/iw
vt1vaZU9qY7E5QyX6Vbrr1RpLZ+Bw2c83+p36adLgRpJHtpfkc3tX6+tdGRDpGtJ2/0sx+PS4MneRI/
M4H4RrKWIx+Gw+VWok+kr+nBZvvc4bfDF9kpYGsZLPyu+97fw4TD6a8X8QkV/eFGscgcdbQIZGpsK
zVocCBpaHcjqTPcz7FfrWOhAwGHAF8ZL34i1Dbb/9v3XOTb4sIenWWdZ17

```
iknDMR2nkO48tfIcdi6lVy9Z2dzxdjcn17O6SjC4nKx73gNqYavcUK9GoRxCVFYjxAM7bT5uSsyMH
RmVwbqykgg9xHCbjYG/GNqUgz1bspKsCi6lTa/DtFoShZgmSbxf4mYlK1WjhqNFRTq1KfSPmdmyMV
JABAGoPOZDaO+O0q98+LqgfLTIpD/ZY+s6v+msTia5ZWoqtWq7dIzmwzsTqAt+JnNvbsBcBUWj0wq
VSmapZMopg6KNTqTYn05yWSdhXdrmfqOS2YkljxYm5J7zxmz/w53hqYWjXWmE+KuDmYE6hANNZiax
OgXVmIVBzJ0l6lHoMPkB1Cm55sdCfUxSd2kNEO9W3a2Or9JVfMFBWkLABUvrYDmdfTlOfYGyHxT1M
tQU6dFA1WqVzWZjZEVb6sxv5AnslfUNppt2quXB5RxqYmpUbvCgIg8v3n9UU3ZWQRKjbmzqdHL0ZY
jg2YgnNz0A4/lKVxNdt2j/7Yv29KmvdqPzmUaNX94jYbv7QLUEN/iQlG145eF/IiZTbBviKp51qh9
WMtt130qLyZD/UCP7ZV7ZW1eoP9RvfWCYPU4GHZNjlsqovAAKPLSZA9h7x9Zt8PR6pJFyeA7PGJrM
lfIot7aWXoh3VP7ZnZrN+EsKJ76v8AZ+kzC0iY3qEdC53WF+k/k/umtwI0mV3VFmqDuT6tNds4aeZ
hAjI7aaz0PYjXw9I/6YHpp+UwSTc7ttfDJ3Fx/uMKugR1LGCYZgGcSYMUeKAF8XgFpHeNeSAcmcO0
NrUKH2rgG18oBLW8Bwgbax/QUS46x+FB+I/pqZ5pXx3TVnS7M6JmrOWFl+VSWOrH5Rc2EklxItnpm
z9tYauctKqpaxOQ3VrDibHUjh6yu3k3gOGZURFZmXMSxNgL2Gg48DPJdv4hqZoOpIb9qQD+FgwYeY
lquILIpOunae8xqKvYTeRbbR3mxdZFSoaacS3RFgGU6oGub3AtccL35SsG28VSz5MQ+RqRQ0yS12Y
g5lufgIAYXHG48RU4CtUqr07Eim71Bh0C6FKZALlu25zCw4WnBt4VKgSjTDFqtTLZeJQC7W/+8401
u3DO4dXmOHOBsnaDU69SnmYIyU2ZbmxcEgG3C9jAwy2QACwHAd0j2dgTUbE1r2FJsMlrdY1A5491s
vvHKWSYorOx17XxN3cli2W4B1N1XQWB115d8ssHscYbC0a9XXFYkO7gqb06J0RNT1SCDa3Ht0jYba
AoKdLsQLDh5mcW2NtVcQytWdbgBKaiygAdg7SfWKUXK2eQJpFTU46Tu2DQZcOj3+Go9Yj+V8p+k4a
ksNg4ovhaFO1sgqi/zFqrNc8uweUJvNDjoanCYhaeHNRzooY//kw21MdUxFZ69U3d2LN3cgO4CwHh
LfbuOui0F4dZ/wC0fn6Sjo4R69VMPTNnqvlB7EQau57lUE+Um2kriSzLfdXA2R8Y462ajhPDhWqDy
IQH8Tcoe2VtRY24lFB5G+b+2WuNq0xlpURalTVadEduRe0/iJux72M5d5QBhrD7rqWPfYj85ygvey
cnZWRjnGkv92KZbDqB2tU8viMyWJxhOi6Dtm33MCnCA5lGVqgNyON7/QxSkmwSJd5UVcFYf5iC/PU
zEETYb01x+zIna1W4H4VDX92EyJH5+g1M6ES33WQ/vj2XpDzs0rNr61qh/GZpN38KaeFVze9VnqAW
+4CUH/gT5zN7S+1f+NvrIRzJM4KvD0m2w/3fEflMVWGhm1oDqHwj+IXuObfEXSl/FU+izLVGM1O+n
Upfxv8AQfpMpn7I5IIvIt91G/eVP4B9f+ZrMAdD/FMhuwbV2HOkfZlmv2f2+MUQkWKNNvum98P4VG
+gMxCCbHc8/unH+p9VH6SVTQUdS/gGOYxlc6AxRRQAtrxXjmDJiK7bOy1xKqGdlCknS1ted/CYnGY
TD4Si+Gw9ZqhqYl6+IqGxzMQAFBGlhYcOUtN+dr5b0Q1kVc1Y89LgHuA185kaFCocKmMrMU6Y3wuG
AF+g/wAyqx1uewCwse3skksrkW9bFexptjcKlXL0fSVqjZrW/d02IvfvtHxlTJQJ+Wkx8wpMq8XmQ
YxQqk9Hh3drC+UFgpJ5CTbx1CtCp3oR5HT85K/pNkXokWzUgmEwSCwAwFNyPxVWZzOPZrX2jhz2JS
xlRvKi1ved+8hCVKdK/Uo4WkP5aVz9ZWYBFOMqBlBybNqsLi9naoqAjkdRIf8AmS+MiqIBoOwSfYN
hgsQ54NtJVPeKVIk/+XvAq0ySbAxYEZdmLbjUx2LqH+Xo0kpfChR95xbVxFlerYDiQo4Dko7uAl4M
ImCwwSwOMqUM+LrkXZM65loofuKARe3H2me2nRNSgyL1iBbvIINvad+0NomqXZgc1RswF+ot72NuJ
4DyMc03JL3Ci7JlRXFlP8J+k79m4c0aFIkqc1AVNDwUk6a9unvOcrfsnPs+sWptTuSqNlRvwXuB5Q
lqhxeTCquWJJ4k3MHD4hqTM9NirtTKM1gT0ZIJAvwvbs/OBiHC6fePBRqT4AawauFrOpAWxI0JYD8
4SlFahFN6HNS3irJUuwVwCdCLG3iJrqGLoYzCVb5lY0X6JbXviFsygkdhsNe+YHE7OrU9WQ2+biPU
cJo91gei/nYjv4Tmpk3Ex5l/u5tpMOr06mbKTmUgXOa1j6i3pOjb2wHua1BSysbsgHxKe0gdombsQ
bEa9oIkdB6lxjMe2JqZ9QigrTXkL39THwOBqYismHpdeo1s3YicXdvwhbk9wg0aByjo6bEnusL87n
slxsnBdDdib1GFmI4BflHdJOWVhWNHtZqd1p0PskRKVHmaaAKCe82v5zCbWW1aoOVRvrN7hMKfhZu
NtBOR9zGxOIulT7Rrlcmo5634eUnFWRFvM8+rcDNvRX4E8ppn/wAJMOFObEYhjY9VUFyOwaGUaYV7
KMtrHgey0ipKTyJNWRU77D9zTP8Aqn3U/pMdeerthVdctRFZTxVgCPeXO52x8OKzWoUtKX+Wp1zDm
JOTsrkYv3Hku7d+n4H7NuzsuJsMCdT5T0Lefd9sRSVaORSr5rWC5hYi2gmU/wCnMRSOtKoe8DMP9t
5CDuORHSM125rfDUH4kPsf0lBR2NiTwo1PMW+s027Oz6tEOagtmy2FwTpfl4zpUa3SK1LyCY8AyuT
FFFFAZa3iBkDV7G0lElcR5RvgxrDFBRd26UIL27SAPQWg4/F51pIFstPD0Kapxy5KYGXTlrNDtnYO
IFZzTpF0ZmZWUrcZjexBN76zC7f2VjRVJrLUpgKOhpK2awJDZmZbjMSBoOFpLeV0yO67WKnDbTKYy
oLfbUzRYdqhaiPc/wBJHnId4K/S1ehJIu1JCLE9Z1N7Dtljs3dnFsTWqKxqtfKO02GgANrACWmxdw
nxFRKlSs+ZXRntb4VQ6AnnFfUlY4N9dpKcZnU3U4h7W7VVVQWi2cy9NiWB/wCxhqYP8VXpCP8AZO/
aG7hrYg/tFNw3S1Gt8QJV2vpbjwGs1C7i0/2dThlyVWZXrZixaoFDBAMxstg3dFnZBle5jTVy3N+F
5DgQ/wCwYVWy2KV3UAEEB6z8dderPQdibodG6mpSVhnUszqhsB2ASz21ulTr1BUWwsgUU7lVCgk6W
4ceElJ5oilkzy5cDoCbm/VAU6+HOVW1MNUZujCNTAt0ha4a3Gyg666az3LY+wzQZTdcqghVBJ4i3b
4mFtnd/DYo5qtMZwLCoLZrcjzHjFKb4jjFHiD4SvXtSoqBmNnqX6qdvnO3/wBEdLUEDAWsrAeOtyL
XnqOG3SRNFqALyFOx+ssq2zejpdHQS+Y/vCxFzbhxnOrU3IuVnJr3LVnWnHelu5K/HQ80w+6wSmFp
AZjYvVY61D3njOmju0FF2ZC3nabpNiVG1d0XuALfpKfGZFcojFgvFsthm5DXXxmBiMXj4p1HBRS7m
/HwSNnD4fCSahvOT+aKzYm76Yh3Q1VU08udFUlsrcCL6WPnNHT3OwYFirnvzW+gE8/xmLqDFVejZl
AWkrWYi/w37OPbOzAbSrqbrWqA/wAZsfI6GbOEqSrUoylk2kZmLpRpVZRjojV4zctONCoR+F9RfxG
o9DLLYmxxSohKq02fMxvYNa/DUjukG7O16lcslSxKqCGAsTrbW2kv7zvK+jKq4mc2ruXhsQS4zU3P
EpbKTzK/paU53Gemf3eR++9j6HQes3givBOw3mZDC7qVjbOUUeOY+QH6zR7O2XToLZB8R6znif0Hd
O28RMbk3qJRSAKzmxOBpVPtKaN3lRf14zqMaRGV6bGww4UU8xf6zopYdE0RFXnlUC/pJzBMd2ABEE
iGYJgAxgxzFAATGMcyMmACvFBvFACYVQTczqWvKwGSU3iAtle8hxWHSoLOoI7OY8D2R6TAiKoRaAG
```

f2vgeidK1BsrqCFuSUdTq1OoOR4g8QfMGCpvEqOP3QAYXqLcZw40Pcwtax+nCXOIphwVPA+3fMftP
CNTfJUFxxU9luYPZMuvWqYWpv23qcteKfZ36q/PjqYejDFQ3L7s1p2r6fXu2eHrq6h0IKngROuk0w
WzMW9A3RtD1kbVW9OB75rsBtKm44hT8pI9uct0sZQq+rNdzyfLyKtbCVqXrRfes10/exbWitI6Zvw
IPnHY2lrMqhwHkT4qmOLr63PtOSvtemvVXN7D9ZXq4qjS9eaXzz5a9DvTw9Wp6sW/vtLBVMhxWNp0
usbt8g1PnylFiNsVX0vlHJdPfjKPaG1lQ5EGer8gOid7n7o7uMyqm19+W5h4bz4vy87GjT2burerS
suzzLPa+3qvBCVdvhoomrM54eP0ke1qIprTDBRV6ENiCNAX7dBoODcJUYPaa4e9QDPimFmruNKY+S
kn3V7zqZzbdxdR0CFiatX4SSdQv3vLsnGtSqqluTm5TqNdytnlyRaoTpOpvQilGmm7+93y77d+Z07
m1cEabviAnTVq7upqJcBCQFAJ0HA8ec4Mfl/aKuQAL0rhQBYAA20EbB7FLK9QXShTQnNxzMBoqjtJ
NoVCkDx8/Geiw8FFJL3ZGDXqb8nJ+80+5NL7R+z4FHuT+U1MrtgYQUsOo7Wux8+HtaWMc3dnJDiKN
eNeIYQMRMEGMTAByY8C8cGACMExzGvGAxgmOYJMAEZHETBvEA5gGOTBJjAUaNeKAEQkqmRLDEQE6V
SIbVbiQiOIErEgMjxWGSquVxcdh7VPMGEIYMjKKkrNXTJRk4u6dmjL47Y9SkbqM6cwNQO8SChUE2E
56+CpvqyKTztY+omJiNjqWdOVux+Zr0drO1qsb9q8vKxSI9hHbEWh7Q2RXGuGdP/jqA+zA/UeczO0
Km0qfXoAD5hTLD1BIlD+kYlZLd5/S5YW0aGrb5fVmnWuDODH7Vo09GcFvkX4m9Bw85jq+KxL6M7W7
VByjzAteQUsO3PyEtUthzftJZcF5vyOE9pwXqR5l5idqVKmgPRp3G9Q/zcF8vWciVABlQWH1PMntM
kweyK9TqUqh78pt6nSaHZ+59U61WVByHxN+gmzQwdKgrRVvEzq2KnVfpO/hyM/QU5hpc3FhxueU1W
yd2GdumxNxfhT4G3I/KPfwl/s3ZFCh9mnxfOdW9ezylhO0qcHJStmtH3nJVZqLjfJ6mZ3sOSgtJFs
rMBYCwCLrbu1t6So2Ds41qgU9RbGofw8vEzdsoIsQCDxBFwZFhsNTpgimiqCbkAWuZ1jLdVkcZK7J
o8UaQAeBHjRgKNHjQAYmNeOYMACJgXiMaABXgtEDHMYERgmE0AxAIwDHMGACijRQAjWSrFFAkgxHE
UUACEO8UUAHBhXiiiAUe0aKFgAekrcVB8RCSgg4Ko8FAiiiQEsQEeKMTCijRRhcKKKKIQo0UUAFGi
ijAUGKKADGIxRQGCY0UUQgYrxooAAYJiijAEwTFFAAYooohH//2Q==">
      <h3>AMAZON</h3>
      FULL STACK DEVELOPER

        Salary:8 LPA

      APPLY NOW

      <img src="https://www.smallbusinesscomputing.com/wp-
content/uploads/2018/07/Freshworks_Logo.jpeg">
      <h3>FRESH WORKS</h3>
      SOFTWARE ENGINEER

        SALARY:5 LPA

      APPLY NOW

      <img src="https://www.the-sun.com/wp-
content/uploads/sites/6/2022/01/kc-google-watching-comp.jpg">
      <h3>GOOGLE</h3>
      TESTER

        SALARY:7 LPA

      APPLY NOW

      <img src="https://1000logos.net/wp-content/uploads/2020/09/Maersk-
Logo.png">
      <h3>MAERSK</h3>
      FULL STACK DEVELOPER

        SALARY:7 LPA

```
      </div>
       <a href="application.html">APPLY NOW</a>
    </div>

    <div tabindex=0 class="card">
      <img
src="https://download.logo.wine/logo/TVS_Motor_Company/TVS_Motor_Company-
Logo.wine.png">
      <h3>TVS</h3>
      <p>MECHANIC</p>
      <div class="link">
        <a href="#">SALARY:3 LPA</a><br>

      </div>
        <a href="apply.html">APPLY NOW</a>

    </div>
<div tabindex=0 class="card">
      <img src="https://mir-s3-cdn-
cf.behance.net/project_modules/max_1200/0af17216851061.562b230c40530.png">
      <h3>ZOHO</h3>
      <p>DEVELOPER</p>
      <div class="link">
        <a href="#">SALARY:4 LPA</a>
      </div>
  <a href="application.html">APPLY NOW</a>
    </div>

  </div>
</div>
    <script>
  window.watsonAssistantChatOptions = {
    integrationID: "a2445f85-b7a3-4c58-980c-72a414ea407b", // The ID of this
integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "1c2b79b4-a6d1-4c4d-bf2f-852aa415e243", // 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>
   </body>
</html>
```

Main.py

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

app = Flask(__name__)

conn = ibm_db.connect(
    "DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURI
TY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=dtk93
338;PWD=honkzW80jUXH3UJN;",
    "", "")
num=[]
num1=[]
num2=[]

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

    return render_template('homepage.html')

@app.route('/login.html')
def login1():
    return render_template('login.html')

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

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

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

@app.route('/login',methods=["GET","POST"])
def login():
     i = int(0)
     if request.method=="POST":
        fn= request.form.get('fn')
        sql = "select * from userdetails"
        stmt = ibm_db.exec_immediate(conn, sql)
        while ibm_db.fetch_row(stmt) != False:
           em=ibm_db.result(stmt,0)
           num1.append(em)
           i=i+1

        for y in range(i):
           ele2=str(fn)
           if ele2==num1[y]:
              return render_template("apply.html")
```

```
@app.route('/register', methods=["GET", "POST"])
def insert():
    i=int(0)
    if request.method == "POST":
        fn = request.form.get('firstname')
        mn = request.form.get('middlename')
        ln = request.form.get('lastname')
        course = request.form.get('course')
        gender = request.form.get('gender')
        #skill = request.form.get('skillset')
        email = request.form.get('email')
        pw = request.form.get('password')
        sql = "select * from userdetails"
        stmt = ibm_db.exec_immediate(conn, sql)
        while ibm_db.fetch_row(stmt) != False:
           em=ibm_db.result(stmt,0)
           num.append(em)
           i=i+1
        for y in range(i):
           ele2=str(fn)
           if ele2==num[y]:
              print("already registered mail")
              return render_template("alertwrong.html")


        insert_sql = "Insert INTO userdetails Values(?,?,?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, fn)
        ibm_db.bind_param(prep_stmt, 2, mn)
        ibm_db.bind_param(prep_stmt, 3, ln)
        ibm_db.bind_param(prep_stmt, 4, course)
        ibm_db.bind_param(prep_stmt, 5,  gender)
        ibm_db.bind_param(prep_stmt, 6, email)
        ibm_db.bind_param(prep_stmt, 7, pw)
        ibm_db.execute(prep_stmt)
        print("inserted")
        return render_template('register.html')

@app.route('/apply',methods=["GET","POST"])
def applyforjob():
    fn = request.form.get('fn')
    ln = request.form.get('ln')
    citizenship = request.form.get('Citizenship')
    city = request.form.get('city')
    twelve = request.form.get('twelth')
    tenth = request.form.get('tenth')
    insert_sql = "insert INTO newapply values(?,?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, fn)
    ibm_db.bind_param(prep_stmt, 2, ln)
    ibm_db.bind_param(prep_stmt, 3, citizenship)
    ibm_db.bind_param(prep_stmt, 4, city)
    ibm_db.bind_param(prep_stmt, 5, twelve)
    ibm_db.bind_param(prep_stmt, 6, tenth)
    ibm_db.execute(prep_stmt)
    return render_template("apply.html")
```

```
if __name__ == '__main__':
    app.run(host='0.0.0.0')
```

13.2 GITHUB & PROJECT DEMO LINK:-

- https://github.com/IBM-EPBL/IBM-Project-21179-1659774640
-