IBM – NALAIYA THIRAN PROJECT

SKILL / JOB RECOMMENDER APPLICATION

INDUSTRY MENTOR	:	KRISHNA CHAITANYA
FACULTY MENTOR	:	KOUSIKA N

TEAM ID	:	PNT2022TMID02811
TEAM LEAD	:	K SIVA SATHWIK
TEAM MEMBER	:	SRINIVAS R
TEAM MEMBER	:	SHASHANK R
TEAM MEMBER	:	SREENATH S

TABLE OF CONTENT

1.INTRODUCTION

1.1 Project overview

So how do you create or edit a profile on our app? Well, it's pretty easy to do because we have a pretty solid wizard to get you started. Some of the things you're going to have to do is upload a photo. You can actually skip this, but it's not recommended because people will engage more with a profile that has a photo on it. The second thing you're going to have to do is write a headline. This is a short phrase that talks about what you do, and it's different than the summary. You want to make this pretty short and sweet, but you also want to make sure that you're writing something that's going to attract people to visit your profile. Now, you can edit this at any time. So, if you change positions or you decide that you want to focus or change your career or make a career change, you can edit the headline at any time. You can also edit things like your headline, for example, if you're looking for a new job, you can put that in there and maybe, you can put the career before you were looking for. So, that way, people can see that you're looking for a new job. we describe the implementation of a skill recommender using the same approach that is commonly used for content-based recommenders. The recommender is based on the identification of skills in user profiles, the calculation of skill similarities, and the recommendation of skills to users. The recommender has been implemented as a web service and is available online.

1.2 Purpose

The Internet-based recruiting platforms become a primary recruitment channel in most companies. While such platforms decrease the recruitment time and advertisement cost, they suffer from an inappropriateness of traditional information retrieval techniques like the Boolean search methods. Consequently, a vast amount of candidates missed the opportunity of recruiting. 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. In order to improve the e-recruiting functionality, many recommender system approaches have been proposed. This article will present a survey of e-recruiting process and existing recommendation approaches for building personalized recommender systems for candidates/job matching.

2. LITERATURE SURVEY

2.1 Existing Problem

- There is no proper resume builder in the existing systems
- The existing website doesn’t provide any good candidate.
- The website is not responsive.
- The website doesn’t provide any learning path to improve my skills.
- It has inaccurate reports of insufficient profiles.

2.2 References

A CAREER PATH RECOMMENDATION FRAMEWORK:

In today's world, recommendation systems are used to solve the problem of information overload in many areas allowing users to focus on important information based on their interests. One of the areas where such systems can play a major role is in helping students achieve their career goals by generating personalized job and skill recommendations.

At present, there are many job posting websites providing a huge amount of information and students need to spend hours to find jobs that match their interests. At the same time, existing job recommendation systems only consider the user's field of interest, but do not take into consideration the user's profile and skills, which can generate more relevant career recommendations for users.

This article was published in March 2017 and authors of this article are: **Bharat Patel, Varun Kakuste, Magdalini Eirinaki**.

JOB RECOMMENDATION BASED ON JOB SEEKER SKILLS: AN EMPIRICAL STUDY

In the last years, job recommender systems have become popular since they successfully reduce information overload by generating personalized job suggestions.

The contributions of this work are two-fold:

- It 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.
- It forwards the proposal of a framework for job recommendation based on professional skills of job seekers.

This article was published in March 2018 and authors of this article are: **Jorge Valverde-Rebaza Ricardo Puma Paul Bustios Nathalia C. Silva**.

RECOMMENDER SYSTEMS: A SURVEY

From the last two decades internet-based recruiting platforms have become a primary channel in most companies for recruiting talents. Such portals decrease the advertisement cost, but they suffer from information overload problem. Job portals using traditional information retrieval techniques such as Boolean search methods are typically using simple word matching algorithms. The main issue of these portals is their inability to understand the complexity of matching between candidates' desires and organizations' requirements. Hence, a vast amount of deserving candidates misses the opportunity to get an appropriate job.

The recent recommender systems have achieved success in e- commerce applications. In order to improve the functionality of e-recruitment process, many recommendation systems approaches have been proposed.

This article was published in March 2019 and authors of this article are: **Juhi Dhameliya, Nikita Desai**.

JOB RECOMMENDATION SYSTEM USING MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING:

This domain is the Hiring process, where a job seeker applies to a job by creating a profile on a job portal by providing all his/her work preferences.

These work preferences of each user can be collected from each user and provide job recommendations based on their preference.

Data acquired for our study has no previous interaction between the user data and Job listing data. With such a dataset, we have addressed the issue of cold start from both User and Job perspective.

This article was published in May 2020 and authors of this article are: **Harsh Jain**.

JOB RECOMMENDATION SYSTEM IN PHP:

This research aims to develop a job web portal for the students in the Faculty of Computer Science and Information Technology (FCSIT), University of Malaya (UM). The main aims of this portal are to connect to the industries and acts as an online recruitment to support the students to find the right IT job after graduation.

Furthermore, this system enhances the understanding concept and importance of the job portal for students in the universities. A survey was conducted to identify the students' problems with the existing portal of the faculty and to gather their requirements which can be incorporated in to the portal to be developed.

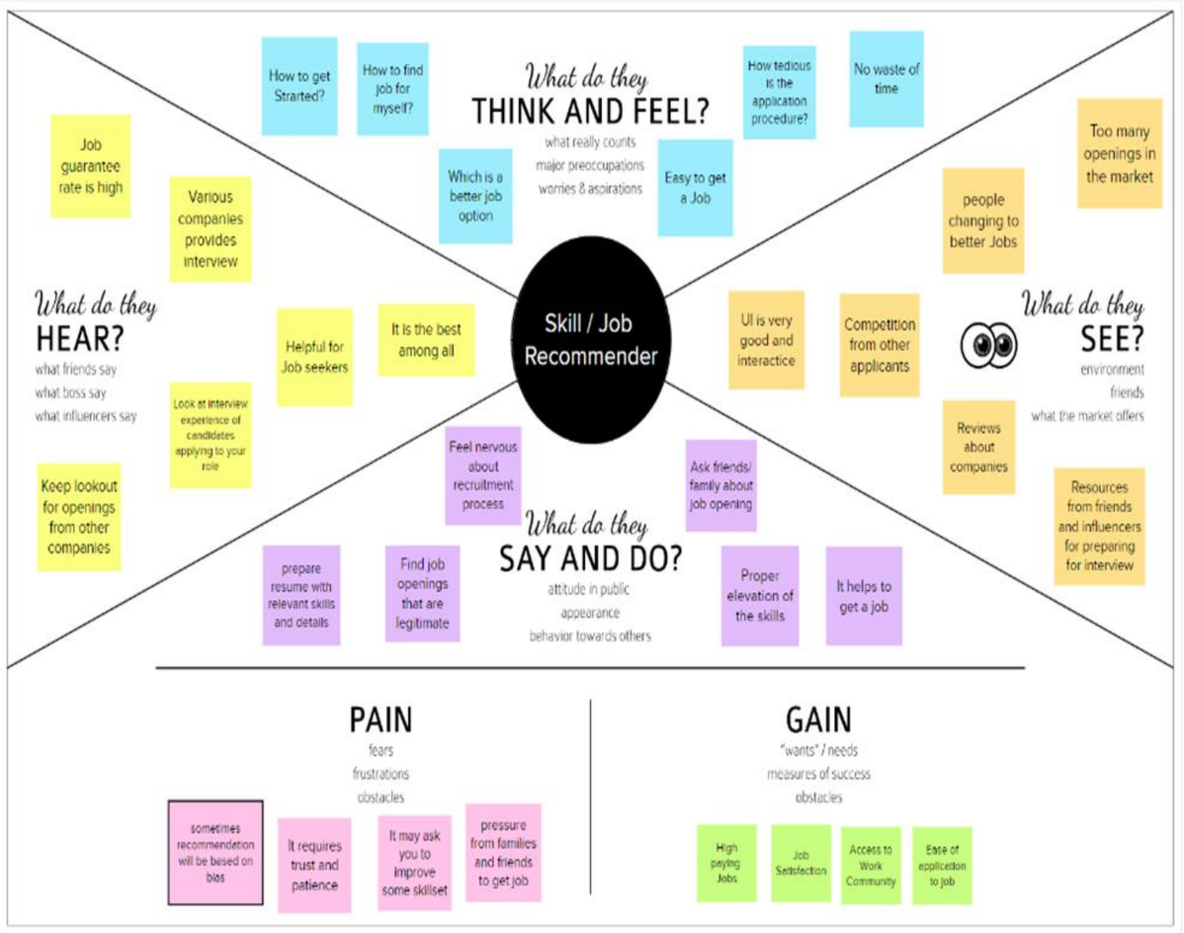
This article was published in Jan 2021 and authors of this article are: **Gupta A, Rothkrantz L**.

2.3 Problem Statement Definition

- Design a skill recommender app that can suggest new skills for users to learn based on their current skillset and career goals.
- The Skill Recommender app will allow users to input their skills and interests, and receive recommendations for other skills they might be interested in. The app will use data from the user's profile and activity to make recommendations.

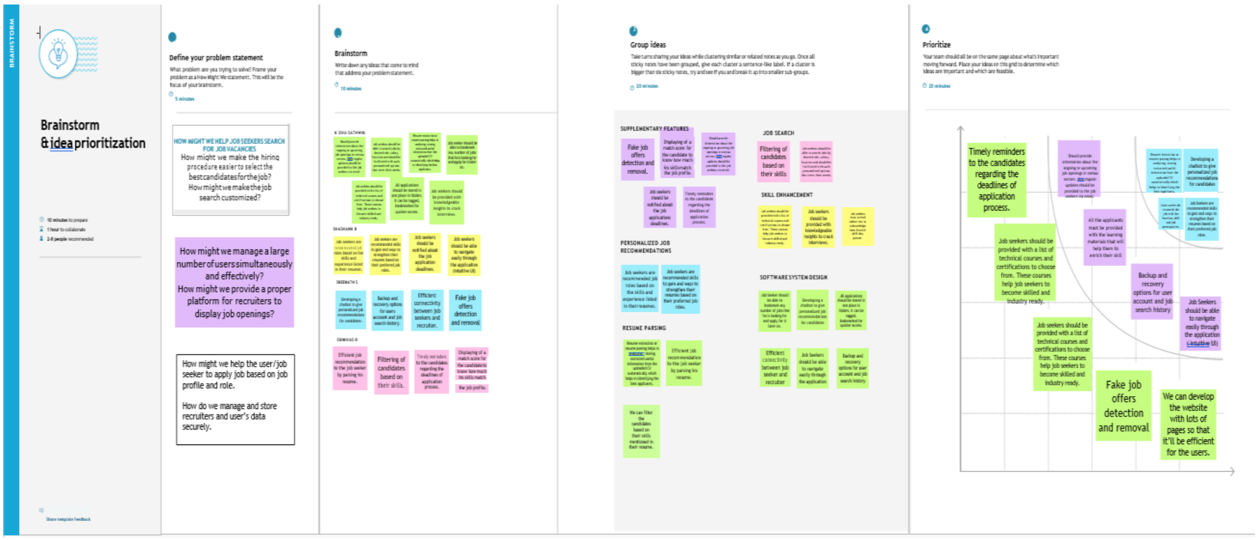
3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
How to get Strarted?
How to find job for myself?
Which is a better job option
Easy to get a Job
How tedious is the application procedure?
No waste of time
What do they
HEAR?
what friends say
what boss say
what influencers say
Job guarantee rate is high
Various companies provides interview
Helpful for Job seekers
It is the best among all
Look at interview experience of candidates applying to your role
Keep lookout for openings from other companies
Skill / Job Recommender
What do they
SEE?
environment
friends
what the market offers
Too many openings in the market
people changing to better Jobs
UI is very good and interactice
Competition from other applicants
Reviews about companies
Resources from friends and influencers for preparing for interview
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Feel nervous about recruitment process
Ask friends/ family about job opening
prepare resume with relevant skills and details
Find job openings that are legitimate
Proper elevation of the skills
It helps to get a job
PAIN
fears
frustrations
obstacles
sometimes recommendation will be based on bias
It requires trust and patience
It may ask you to improve some skillset
pressure from families and friends to get job
GAIN
"wants" / needs
measures of success
obstacles
High paying Jobs
Job Satisfaction
Access to Work Community
Ease of application to job

3.2 Ideation & Brainstorming



BRAINSTORM

Brainstorm & idea prioritization

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

HOW MIGHT WE HELP JOB SEEKERS SEARCH FOR JOB VACANCIES
How might we make the hiring procedure easier to select the best candidates for the job?
How might we make the job search customized?

How might we manage a large number of users simultaneously and effectively?
How might we provide a proper platform for recruiters to display job openings?

How might we help the user/job seeker to apply job based on job profile and role.

How do we manage and store recruiters and user's data securely.

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

K SIVA SATHWIK

All applications should be stored in one place in folders. It can be tagged, bookmarked for quicker access.

Job seekers should be provided with knowledgeable insights to crack interviews.

SHASHANK R

Job seekers are recommended job roles based on the skills and experience listed in their resumes.

Job seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.

Job seekers should be notified about the job application deadlines.

Job seekers should be able to navigate easily through the application (intuitive UI)

SREENATH S

Developing a chatbot to give personalized job recommendations for candidates.

Backup and recovery options for users account and job search history.

Efficient connectivity between job seekers and recruiter.

Fake job offers detection and removal

SRINIVAS R

Efficient job recommendation to the job seeker by parsing his resume.

Filtering of candidates based on their skills.

Timely reminders to the candidates regarding the deadlines of application process.

Displaying of a match score for the candidate to know how much his skills match the job profile.

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

SUPPLEMENTARY FEATURES

Fake job offers detection and removal.

Displaying of a match score for the candidate to know how much his skills match the job profile.

Job seekers should be notified about the job applications deadlines.

Timely reminders to the candidates regarding the deadlines of application process.

JOB SEARCH

Filtering of candidates based on their skills

SKILL ENHANCEMENT

Job seekers should be provided with knowledgeable insights to crack interviews.

PERSONALIZED JOB RECOMMENDATIONS

Job seekers are recommended job roles based on the skills and experience listed in their resumes.

Job seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.

RESUME PARSING

Efficient job recommendation to the job seeker by parsing his resume.

We can filter the candidates based on their skills mentioned in their resume.

SOFTWARE SYSTEM DESIGN

Job Seeker should be able to bookmark any number of jobs that he is looking for and apply for it later on.

Developing a chatbot to give personalized job recommendations for candidates

All applications should be stored in one place in folders. It can be tagged, bookmarked for quicker access.

Efficient connectivity between job seeker and recruiter

Job Seekers should be able to navigate easily through the application

Backup and recovery options for user account and job search history

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Timely reminders to the candidates regarding the deadlines of application process.

Job seekers should be provided with a list of technical courses and certifications to choose from. These courses help job seekers to become skilled and industry ready.

All the applicants must be provided with the learning materials that will help them to enrich their skill

Developing a chatbot to give personalized job recommendations for candidates

Job Seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.

Backup and recovery options for user account and job search history

Job Seekers should be able to navigate easily through the application (intuitive UI)

Job seekers should be provided with a list of technical courses and certifications to choose from. These courses help job seekers to become skilled and industry ready.

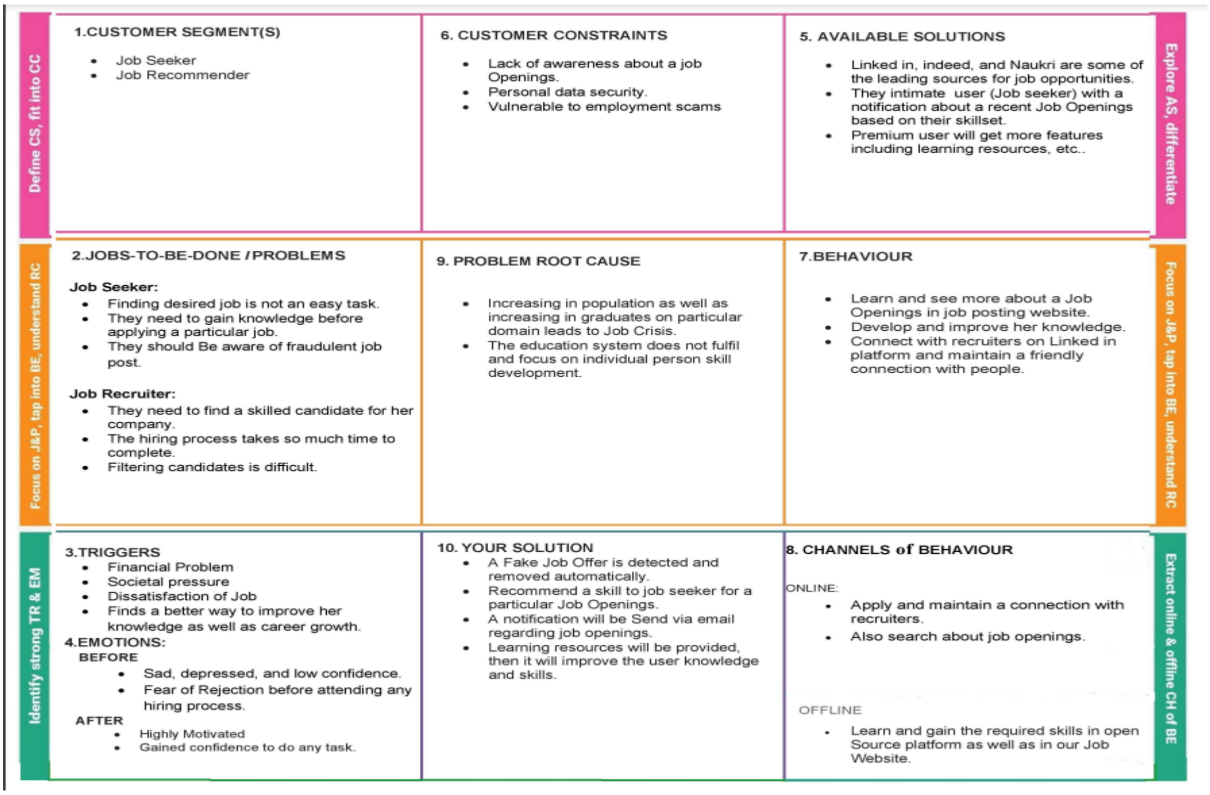
Fake job offers detection and removal

We can develop the website with lots of pages so that it'll be efficient for the users.

3.3 Proposed Solution

Premium policy is an issue to the users. Look for field-based jobs as searching for fields as a whole is time-consuming. Salary calculator for the estimation of the pay. Earnings estimator based on knowledge of users.Open doors for every users as there is free accessProviding recruiters with better candidatesData can be scaled up and scaled down according to the number of current job openings available.

3.4 Problem Solution fit



Define CS, fit into CC

1.CUSTOMER SEGMENT(S)

- Job Seeker
- Job Recommender

6. CUSTOMER CONSTRAINTS

- Lack of awareness about a job Openings.
- Personal data security.
- Vulnerable to employment scams

5. AVAILABLE SOLUTIONS

- Linked in, indeed, and Naukri are some of the leading sources for job opportunities.
- They intimate user (Job seeker) with a notification about a recent Job Openings based on their skillset.
- Premium user will get more features including learning resources, etc..

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2.JOBS-TO-BE-DONE / PROBLEMS

Job Seeker:

- Finding desired job is not an easy task.
- They need to gain knowledge before applying a particular job.
- They should Be aware of fraudulent job post.

Job Recruiter:

- They need to find a skilled candidate for her company.
- The hiring process takes so much time to complete.
- Filtering candidates is difficult.

9. PROBLEM ROOT CAUSE

- Increasing in population as well as increasing in graduates on particular domain leads to Job Crisis.
- The education system does not fulfil and focus on individual person skill development.

7.BEHAVIOUR

- Learn and see more about a Job Openings in job posting website.
- Develop and improve her knowledge.
- Connect with recruiters on Linked in platform and maintain a friendly connection with people.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3.TRIGGERS

- Financial Problem
- Societal pressure
- Dissatisfaction of Job
- Finds a better way to improve her knowledge as well as career growth.

4.EMOTIONS:

BEFORE

- Sad, depressed, and low confidence.
- Fear of Rejection before attending any hiring process.

AFTER

- Highly Motivated
- Gained confidence to do any task.

10. YOUR SOLUTION

- A Fake Job Offer is detected and removed automatically.
- Recommend a skill to job seeker for a particular Job Openings.
- A notification will be Send via email regarding job openings.
- Learning resources will be provided, then it will improve the user knowledge and skills.

8. CHANNELS of BEHAVIOUR

ONLINE:

- Apply and maintain a connection with recruiters.
- Also search about job openings.

OFFLINE

- Learn and gain the required skills in open Source platform as well as in our Job Website.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional Requirement (Epic)	**Sub Requirement (Story / Sub-Task)**
Collect DataSet	Data from different sources are collected in order to get optimized result
Data Cleaning	When combining data from multiple sources there are duplicated data and hence we clean the data 1st

Data modelling	Identify the relationship between various parameters.
Prediction and analysis	The length of stay is predicted with the Machine learning algorithm

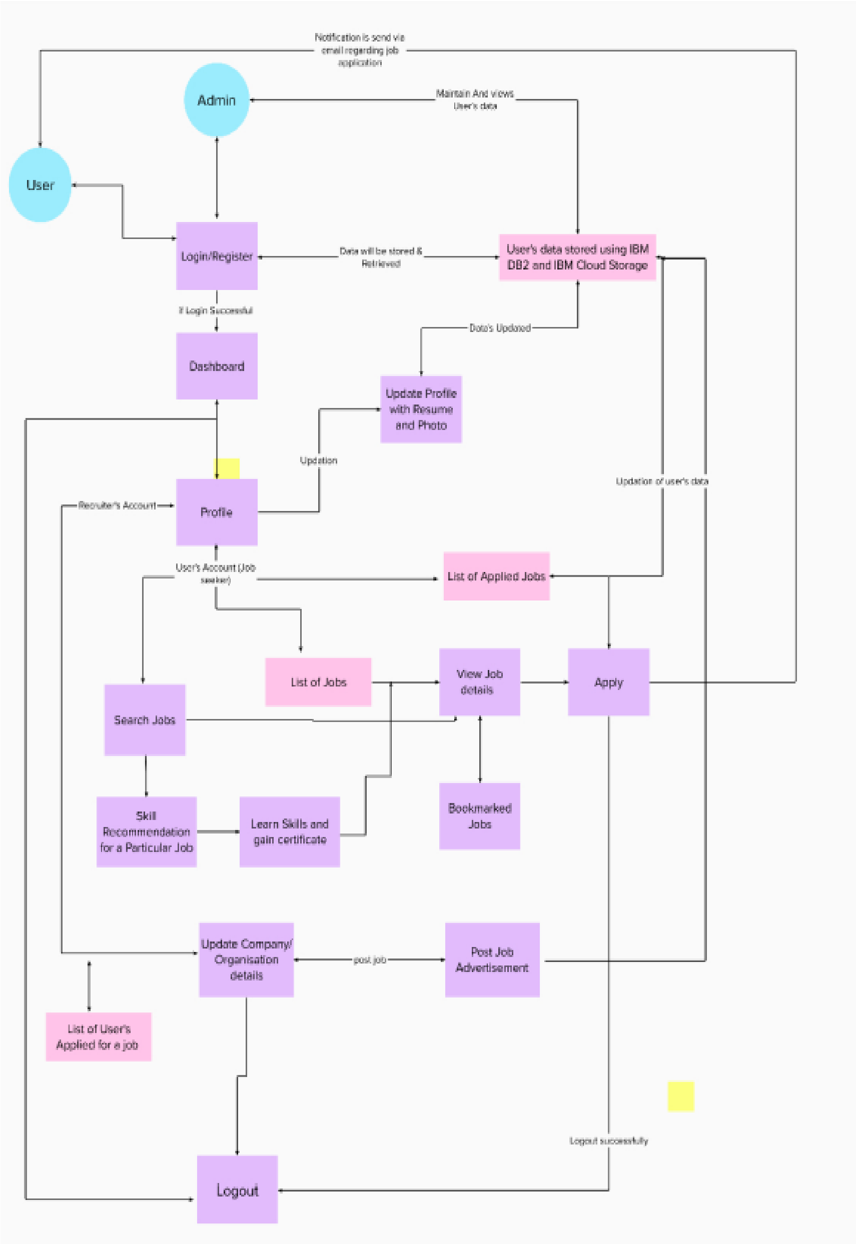
4.2 Non-Functional requirements

Non-Functional Requirement	**Description**
Usability	User can view and visualise the data through the interactive dashboard and predict the length of stay of patients with machine learning algorithm
Security	IBM Cognos provides better security. The dataset uploaded to the dashboard cannot be downloaded or accessed by external sources
Reliability	The dashboard and the prediction is very reliable and provide prediction with more accuracy
Performance	The length of stay of patients is predicted with more accuracy
Availability	The predicted length of stay and the visualization will be available in cognos analysis

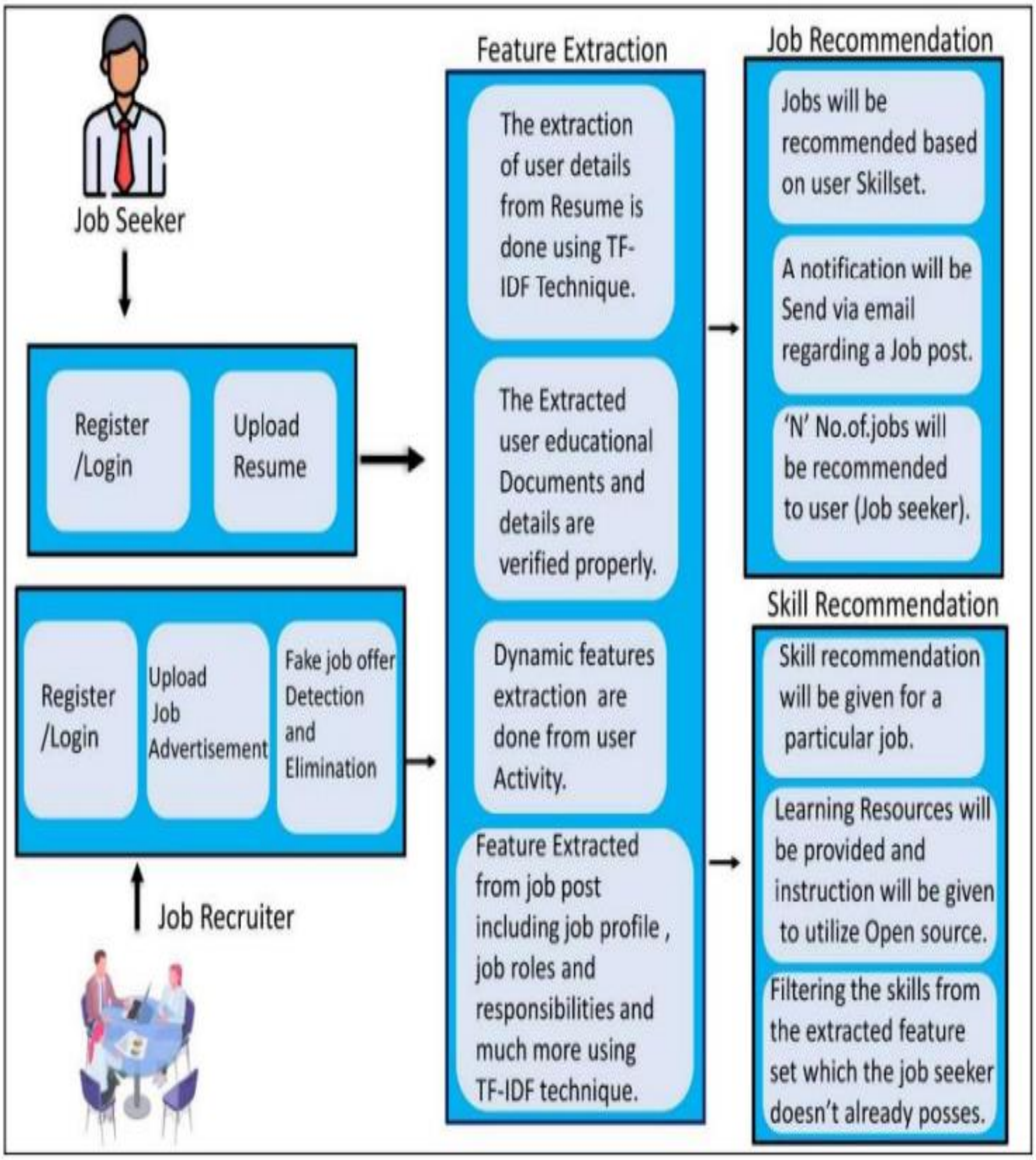
Scalability	The software is scalable and extendable. Because it allow multiple user to handle the data at the same time

5. PROJECT DESIGN

5.1 Data Flow Diagrams


Notification is send via email regarding job application
Admin
Maintain And views User's data
User
Login/Register
Data will be stored & Retrieved
User's data stored using IBM DB2 and IBM Cloud Storage
If Login Successful
Dashboard
Data's Updated
Update Profile with Resume and Photo
Updation
Updation of user's data
Recruiter's Account
Profile
User's Account (Job seeker)
List of Applied Jobs
List of Jobs
View Job details
Apply
Search Jobs
Skill Recommendation for a Particular Job
Learn Skills and gain certificate
Bookmarked Jobs
Update Company/ Organisation details
post job
Post Job Advertisement
List of User's Applied for a job
Logout successfully
Logout

5.2 Solution & Technical Architecture


Job Seeker
Register /Login
Upload Resume
Register /Login
Upload Job Advertisement
Fake job offer Detection and Elimination
Job Recruiter
Feature Extraction
The extraction of user details from Resume is done using TF-IDF Technique.
The Extracted user educational Documents and details are verified properly.
Dynamic features extraction are done from user Activity.
Feature Extracted from job post including job profile , job roles and responsibilities and much more using TF-IDF technique.
Job Recommendation
Jobs will be recommended based on user Skillset.
A notification will be Send via email regarding a Job post.
'N' No.of.jobs will be recommended to user (Job seeker).
Skill Recommendation
Skill recommendation will be given for a particular job.
Learning Resources will be provided and instruction will be given to utilize Open source.
Filtering the skills from the extracted feature set which the job seeker doesn't already posses.

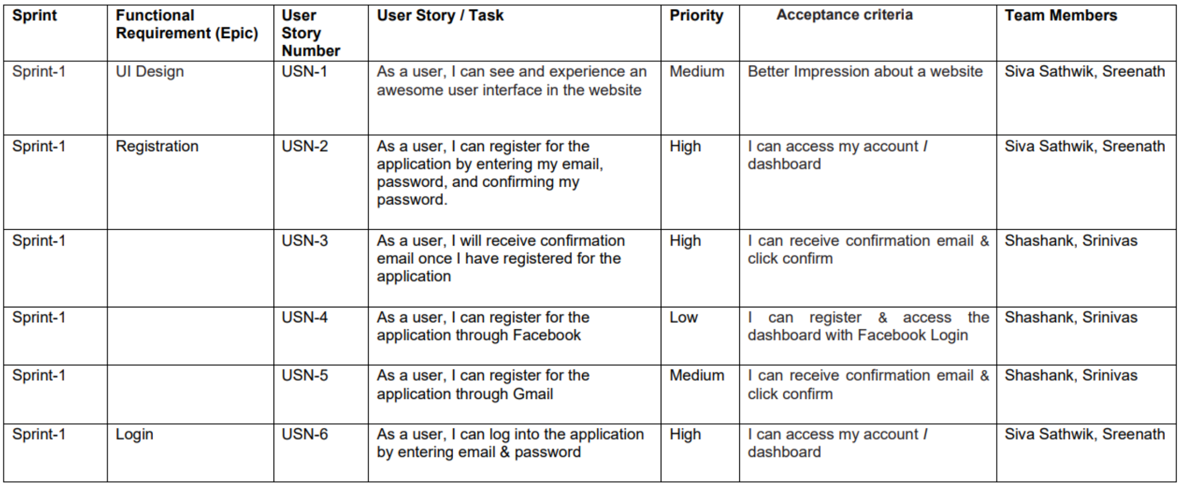
5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Dashboard	USN-1	As a user, I can upload the dataset to the dashboard	I can access dashboard	High	Sprint-1
	View	USN-2	As a user, I can view the patient details	I can visualize the data	medium	Sprint-2
Admin	Analyse	USN-3	As a user, I will analyse the given dataset	I can analyse the dataset	High	Sprint-3
	Predict	USN-4	As a user, I will predict the length of stay	I can predict the length of stay	High	Sprint-4
	Collect data	USN-5	As a analyst I need to collect the dataset		High	Sprint-1
	Prepare data	USN-6	As an analyst I need to do	I can extract the	High	Sprint-2

			feature extraction	parameters that have impact the length of stay		
Visualization	Dashboard	USN-7	As a user I can prepare data by using visualization technique	I can prepare the data with visualization technique	Medium	sprint -2

6. PROJECT PLANNING

6.1 Sprint Planning & Estimation

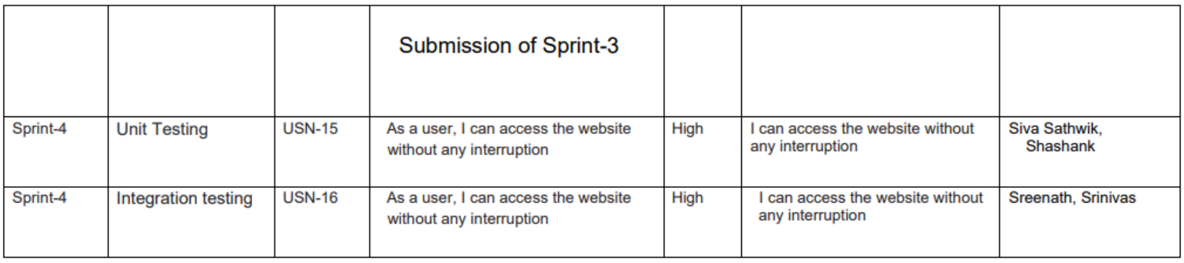


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-1	UI Design	USN-1	As a user, I can see and experience an awesome user interface in the website	Medium	Better Impression about a website	Siva Sathwik, Sreenath
Sprint-1	Registration	USN-2	As a user, I can register for the application by entering my email, password, and confirming my password.	High	I can access my account / dashboard	Siva Sathwik, Sreenath
Sprint-1		USN-3	As a user, I will receive confirmation email once I have registered for the application	High	I can receive confirmation email & click confirm	Shashank, Srinivas
Sprint-1		USN-4	As a user, I can register for the application through Facebook	Low	I can register & access the dashboard with Facebook Login	Shashank, Srinivas
Sprint-1		USN-5	As a user, I can register for the application through Gmail	Medium	I can receive confirmation email & click confirm	Shashank, Srinivas
Sprint-1	Login	USN-6	As a user, I can log into the application by entering email & password	High	I can access my account / dashboard	Siva Sathwik, Sreenath

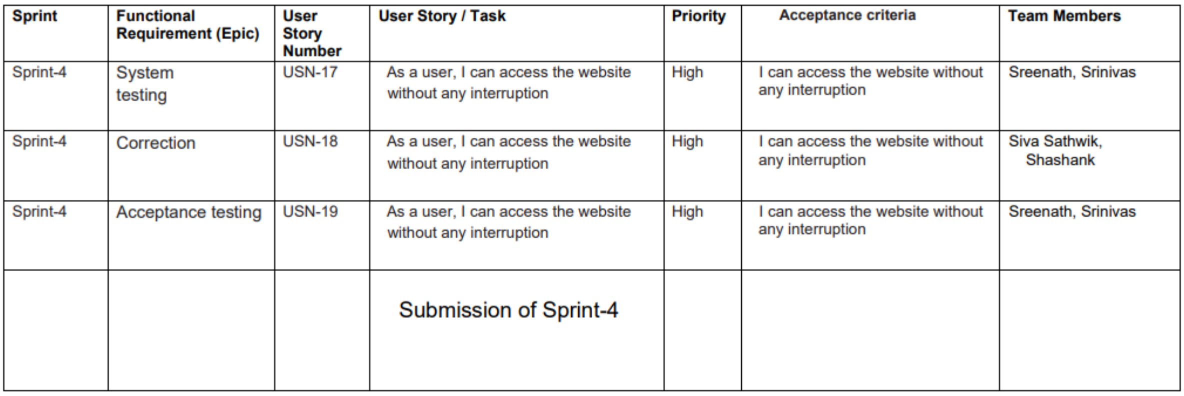
Sprint-!	Flask	USN-7	As a user, I can access the website in a second	High	I can access my account / dashboard	Shashank, Srinivas

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-1	Dashboard	USN-8	As a user, If I Logged in correctly, I can view my dashboard and I can navigate to any pages which are already listed there.	High	I can access all the pages/ dashboard	Siva Sathwik, Sreenath
			Submission Of Sprint-1			
Sprint-2	User Profile	USN-9	As a user, I can view and update my details	Medium	I can modify my details/data	Siva Sathwik, Sreenath
Sprint-2	Database	USN-10	As a user, I can store my details and data in the website w	Medium	I can store my data	Siva Sathwik, Sreenath
Sprint-2	Cloud Storage	USN-11	As a user, I can upload my photo, resume and much more in the website.	Medium	I can Upload my documents and details	Shashank, Srinivas
Sprint-2	Chatbot	USN-12	As a user, I can ask the Chatbot about latest job openings, which will help me and show the recent job openings based on my profile	High	I can know the recent job openings	Shashank, Srinivas
Sprint-2	Identity-Aware	USN-13	As a User, I can access my account by entering by correct login credentials. My user credentials is only displayed to me.	High	I can have my account safely	Shashank, Srinivas
			Submission of Sprint-2			

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-3	Sendgrid service	USN-14	As a user, I can get a notification or mail about a job opening with the help of sendgrid service.	Medium	I can get a notification in a second.	Shashank, Srinivas
Sprint-3	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.	High	I can gain the knowledge and skills	Shashank, Srinivas
Sprint-3	Docker	USN-16	As a user, I can access the website in any device	High	I can access my account in any device	Shashank,Sreenath
Sprint-3	Kubernates	USN-17	As a user, I can access the website in any device	High	I can access my account in any device	Shashank,Sreenath
Sprint-3	Deployment in cloud	USN-18	As a user, I can access the website in any device	High	I can access my account in any device	Siva Sathwik, Shashank
Sprint-3	Technical support	USN-19	As a user, I can get a customer care support from the website which will solve my queries.	Medium	I can tackle my problem & queries.	Siva Sathwik, Shashank



			Submission of Sprint-3			
Sprint-4	Unit Testing	USN-15	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Siva Sathwik, Shashank
Sprint-4	Integration testing	USN-16	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Sreenath, Srinivas



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-4	System testing	USN-17	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Sreenath, Srinivas
Sprint-4	Correction	USN-18	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Siva Sathwik, Shashank
Sprint-4	Acceptance testing	USN-19	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Sreenath, Srinivas
			Submission of Sprint-4			

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

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

7.1 Feature 1

```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "1df8b62f-ee5c-42bb-9a2f-6edcb89d85f7", // The ID of this integration.
    region: "jp-tok", // The region your integration is hosted in.
    serviceInstanceID: "ee519db2-aba8-4373-a6a2-e4e6fea336d7", // The ID of your service instance.
    onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });
 </script>
```

7.2 Feature 2

Job_post.html

```
<!DOCTYPE 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>Document</title>
    <link rel="stylesheet" href="static/css/job_post.css" />
  </head>
    <body style="background-color: rgb(255, 204, 0)" height="40%" width="80%">
    <script>
      window.watsonAssistantChatOptions = {
        integrationID: "6e54406e-1b6b-4b7c-8b9e-a21f9adbb3de", // The ID of this integration.
        region: "au-syd", // The region your integration is hosted in.
          serviceInstanceID: "bc1d370b-fe79-4e14-be04-a0cacaef09e2", // 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>
    <header>
      <div class="wrapper">
        <div class="logo">
          <a href="/"><img src="static/img/job_logo.png" alt="" /></a>
        </div>

        <ul class="nav-area">
          <li><a href="posts">post Jobs</a></li>
          <li><a href="list">view job</a></li>
        </ul>
      </div>

      <section>
        <div class="contentBx">
          <div class="card">
            <div class="formBx">
              <h2>Job Posting form</h2>
```

```
<br />
<form action="{{ url_for('addrec') }}" method="get|post">
  <div class="inputBx">
    <span>Company name</span>
    <input
      type="text"
      name="dname"
      id="dname"
      placeholder=""
      required
      class="form-control"
    />
  </div>
  <br />

  <div class="inputBx">
    <span>Job title</span>
    <input
      type="text"
      name="dtitle"
      id="dtitle"
      placeholder=""
      required
      class="form-control"
    />
  </div>
```

```
<br />
<div class="inputBx">
 <span>Job role</span>
 <input
  type="text"
  name="drole"
  id="drole"
  placeholder=""
  required
  class="form-control"
 />
</div>
<br />
<div class="inputBx">
 <span>Job Description</span>
 <label for="Description"></label>
 <textarea
  rows=""
  cols=""
  name="description"
  id="description"
 ></textarea>
</div>
<br />
<div class="remember">
 <input type="checkbox" id="rem1" name="" value="" />
```

```
         <label for="rem1"
          >I agree to the terms and condition of the form</label
         ><br />
        </div>
        <br />
        <div class="inputBx">
         <input type="submit" value="Submit" name="" />
        </div>
       </form>
      </div>
     </div>
    </div>
   </section>
  </header>

  <div class="grid-container">
   <div class="grid">
    <img
     src="../static/img/33172-01-finishig-studies.gif"
     width="100px"
     height="100px"
    />
       <h5>Get your dream job within a week.<br />Gain on-demand skills.</h5>
   </div>
   <div class="grid">
```

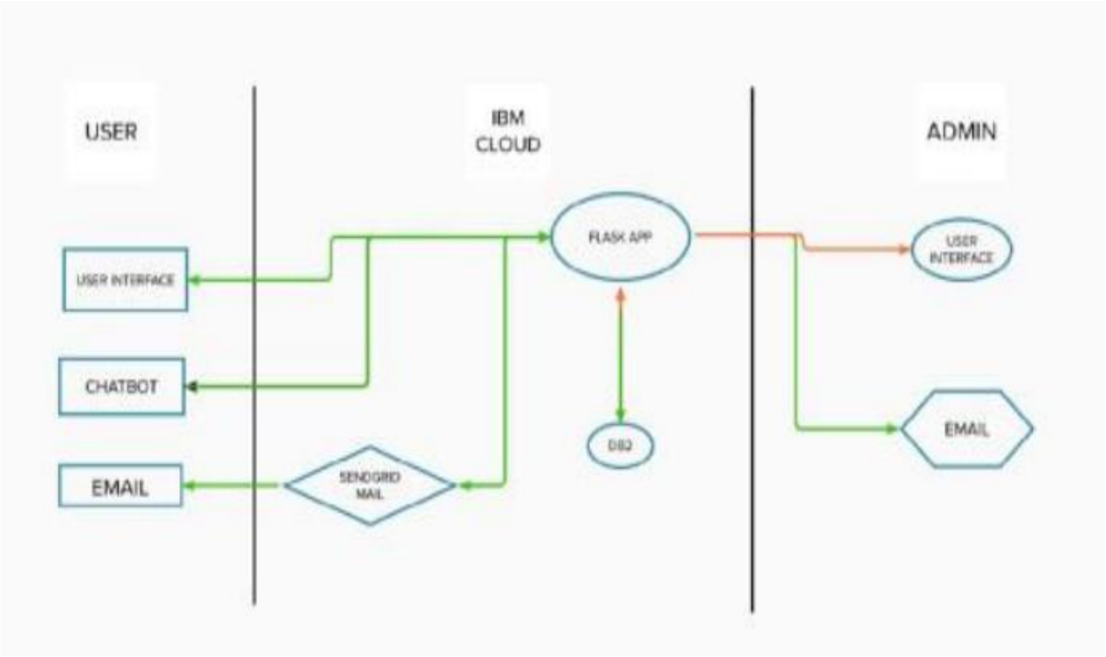
```
                    <img      src="../static/img/27425-train.gif"      width="100px"
height="100px" />
        <h5>Personalized Recommendation.</h5>
      </div>
      <div class="grid">
        <img
          src="../static/img/75164-glurtr-anmation.gif"
          width="100px"
          height="100px"
        />
        <h5>Learn at your own pace, with lifetime access.</h5>
      </div>
    </div>

    <hr id="start" />
    <div class="job-sign">
      <br /><br />
      <h1>Find the best job for your career</h1>
      <button id="sign-center">View jobs</button>
    </div>
  </body>
</html>
```

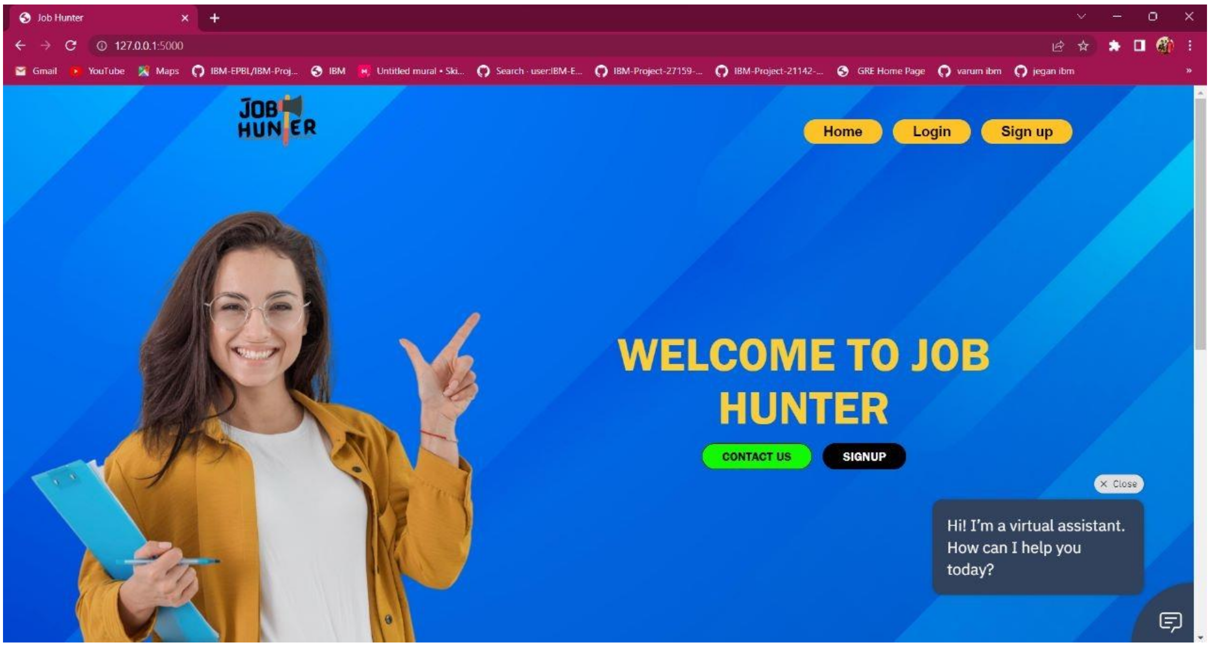
8. TESTING

Black Box testing is the method that does not consider the internal structure, design, and product implementation to be tested. In other words, the tester does not know its internal functioning. The Black Box only evaluates the external behaviour of the system. The inputs received by the system and the outputs on responses it produces are tested. The White Box test method is the one that looks at the code and structure of the product to be tested and uses that knowledge to perform the tests. This method is used in the Unit testing phase, although it can also occur in other stages such as Integration tests. For the execution of this method, the tester or the person who will use this method must have extensive knowledge of the technology used to develop the program.

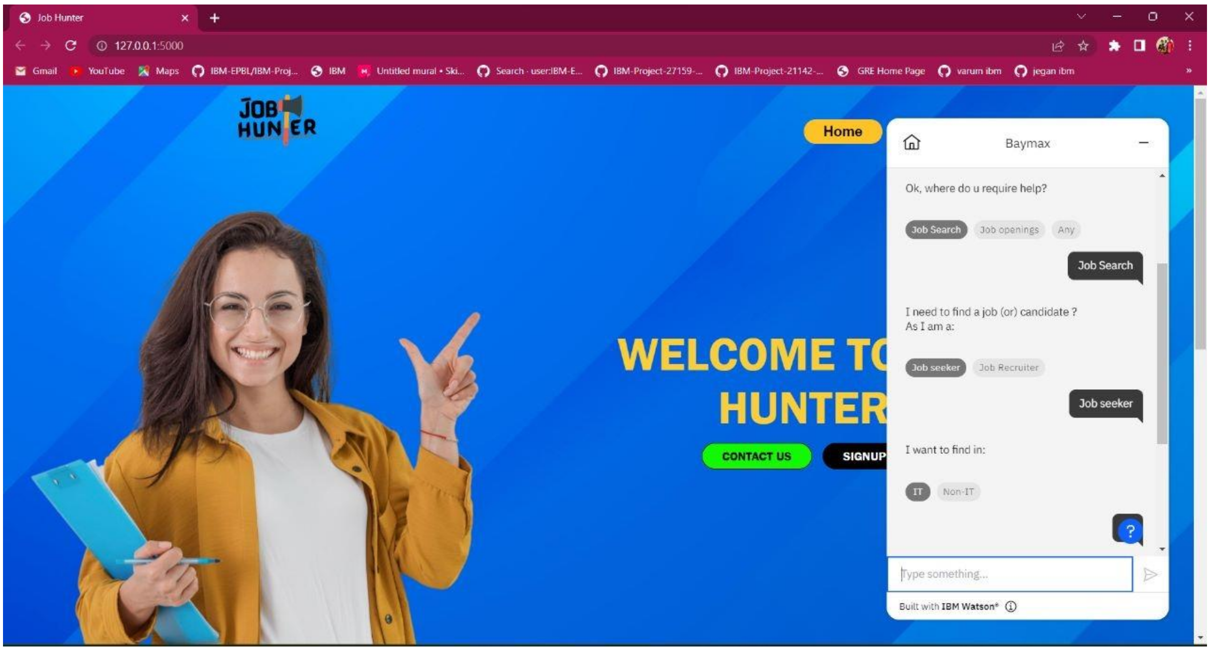
8.1. Test Cases



9. RESULTS



Hi! I'm a virtual assistant. How can I help you today?



Ok, where do u require help?

Job Search | Job openings | Any

Job Search

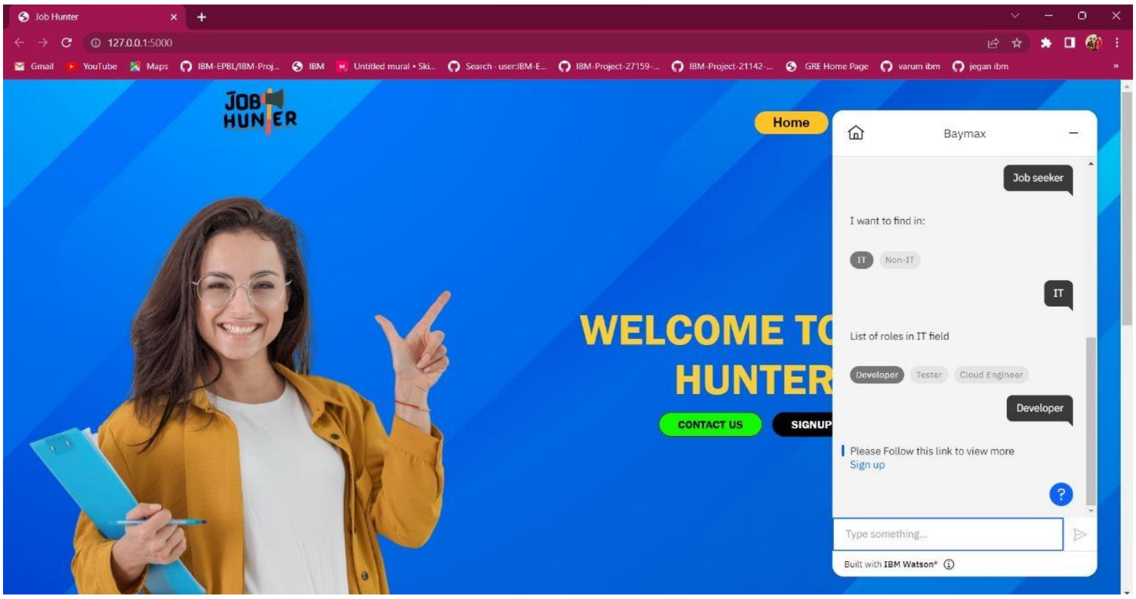
I need to find a job (or) candidate ?
As I am a:

Job seeker | Job Recruiter

Job seeker

I want to find in:

IT | Non-IT



Job seeker

I want to find in:

IT Non-IT

IT

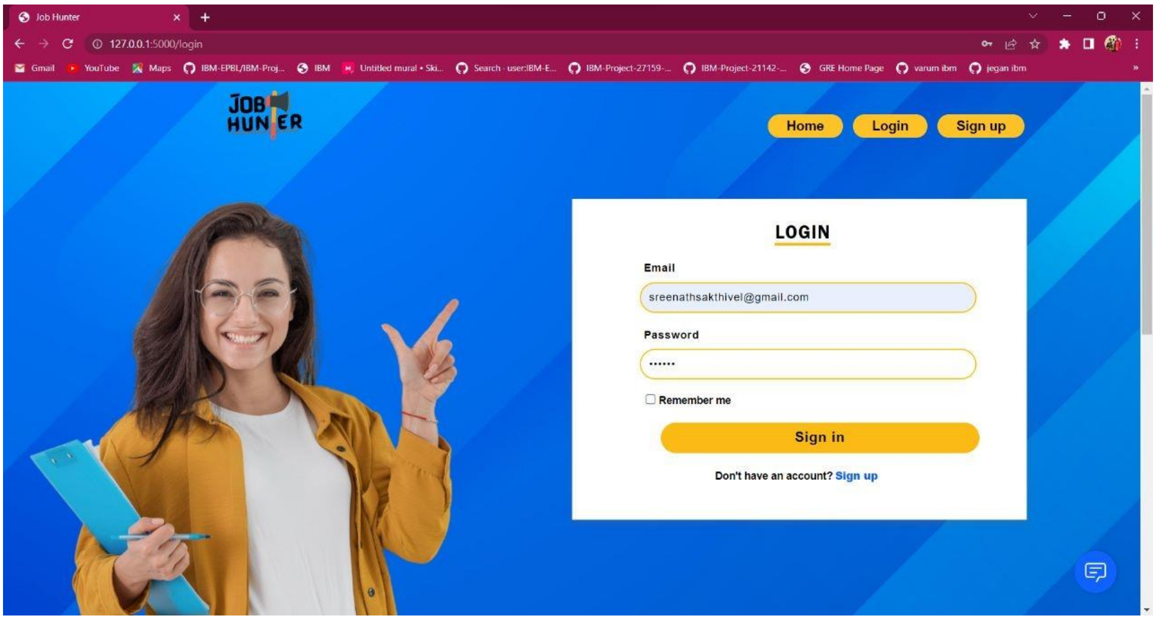
List of roles in IT field

Developer Tester Cloud Engineer

Developer

Please Follow this link to view more
Sign up

Type something...



LOGIN

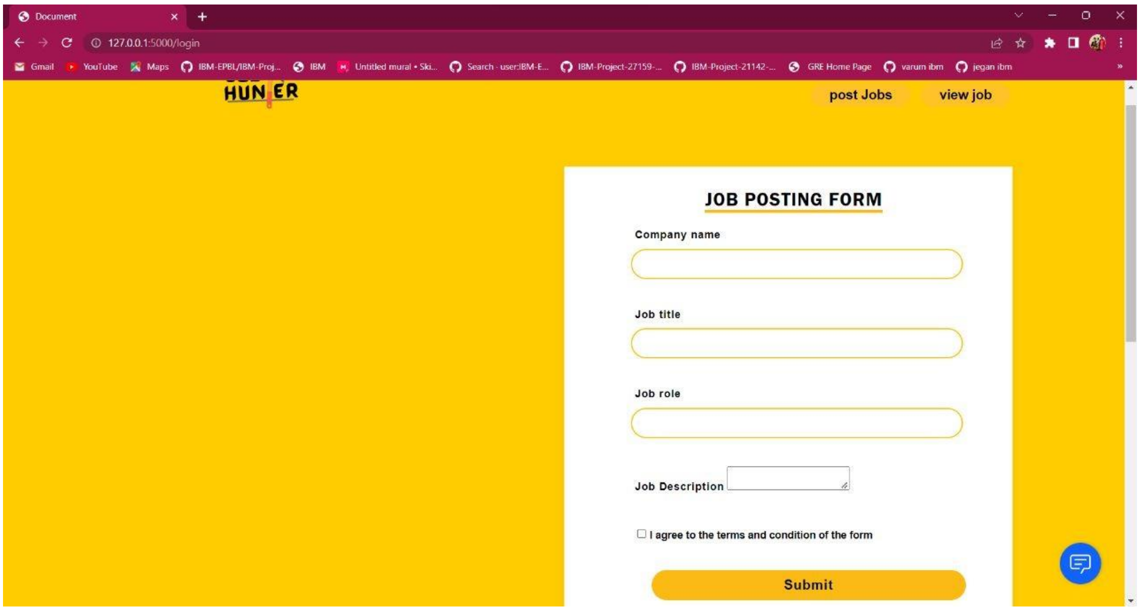
Email
sreenathsakthivel@gmail.com

Password
••••••

□ Remember me

Sign in

Don't have an account? Sign up



JOB POSTING FORM

Company name

Job title

Job role

Job Description

□ I agree to the terms and condition of the form

Submit

9.1 Performance Metrics

The performance of a recommendation algorithm is evaluated by using some specific metrics that indicate the accuracy of the system. The type of metric used depends on the type of filtering technique. Root Mean Square Error (RMSE), Receiver Operating Characteristics (ROC), Area Under Cover (AUC), Precision, Recall and F1 score is generally used to evaluate the performance or accuracy of the recommendation algorithms.

Root-mean square error (RMSE). RMSE is widely used in evaluating and comparing the performance of a recommendation system model compared to other models. A lower RMSE value indicates higher performance by the recommendation model. RMSE, as mentioned by [69], can be as represented as follows:

RMSE=√ 1/Np∑u,i(pui−rui)^2 (1)

where, N_p is the total number of predictions, p_{ui} is the predicted
rating that a user u will select an item i and r_{ui} is the real rating.

Precision. Precision can be defined as the fraction of correct recommendations or predictions (known as True Positive) to the total number of recommendations provided, which can be as represented as follows:

Precision=True Positive (TP)/True Positive(TP)+False Positive (FP)(2)

It is also defined as the ratio of the number of relevant recommended items to the number of recommended items expressed as percentages.

Recall. Recall can be defined as the fraction of correct recommendations or predictions (known as True Positive) to the total number of correct relevant recommendations provided, which can be as represented as follows:

Recall=True Positive (TP)/True Positive(TP)+False Negative (FN) (3)

It is also defined as the ratio of the number of relevant recommended items to the total number of relevant items expressed as percentages.

F1 Score. F1 score is an indicator of the accuracy of the model and ranges from 0 to 1, where a value close to 1 represents higher recommendation or prediction accuracy. It represents precision and recall as a single metric and can be as represented as follows:

$F1\ score=2\times Precision*Recall/Precision+Recall$ (4)

Coverage. Coverage is used to measure the percentage of items which are recommended by the algorithm among all of the items.

Accuracy. Accuracy can be defined as the ratio of the number of total correct recommendations to the total recommendations provided, which can be as represented as follows:

$Accuracy=TP+FN/TP+FN+TN+FP$ (5)

Intersection over union (IoU). It represents the accuracy of an object detector used on a specific dataset [70].

$IoU=TP/TP+FN+FP$ (6)

ROC. ROC curve is used to conduct a comprehensive assessment of the algorithm's

performance [57].

AUC. AUC measures the performance of recommendation and its baselines as well as the quality of the ranking based on pairwise comparisons [5].

Rank aware top-N metrics. The rank aware top-N recommendation metric finds some of the interesting and unknown items that are presumed to be most attractive to a user [71]. Mean reciprocal rank (MRR), mean average precision (MAP) and normalized discounted cumulative gain (NDCG) are three most popular rank aware metrics.

MRR. MRR is calculated as a mean of the reciprocal of the position or rank of first relevant recommendation [72,73]. MRR as mentioned by [72,73] can be expressed as follows:

$MRR=1/Nu \sum u\in Nu\ 1/Lnu[k]\in Ru$ (7)

where u, N_u and R_u indicate specific user, total number of users and the set of items rated by the user, respectively. L indicates list of ranking length (n) for user (u) and k represents the position of the item found in the he lists L.

MAP: MAP is calculated by determining the mean of average precision at the points where relevant products or items are found. MAP as mentioned by [73] can be expressed as follows.

MAP=1Nu|Ru|∑nk=1k(Lnu[k]∈Ru)Pu@k (8)

where Pu represents precision in selecting relevant item for the user.

NDCG: NDCG is calculated by determining the graded relevance and positional information of the recommended items, which can be expressed as follows [73].

NDCGu=∑nk=1G(u,n,k)D(k)/∑nk=1G∗(u,n,k)D(k) (9)

where D (k) is a discounting function, G (u, n, k) is the gain obtained recommending an item found at k-th position from the list L and G^* (u, n, k) is the gain related to kth item in the ideal ranking of n size for u user.

10. ADVANTAGES & DISADVANTAGES

Advantages

- 1. Job recommendations can help you find a job that fits your skills and interests.

- Job recommendations can save you time by showing you jobs that are more likely to be a good match for you.

- Job recommendations can help you discover new opportunities that you may not have considered otherwise.

- The biggest advantage of job recommendation systems for recruiters is that they can help save time by automating the process of finding and recommending candidates for open positions.

- Job recommendation systems can also provide a more objective perspective on candidates by using data and algorithms to identify which candidates are the best match for a specific job.

- In addition, job recommendation systems can help recruiters keep track of a large pool of candidates and automatically notify them when new job openings that match their skills and experience become available.

- Job recommendations can help you find jobs that are not advertised to the general public.

- Job recommendations can help you connect with recruiters and hiring managers.

- Job recommendations can help you find jobs that match your salary requirements.

- Job recommendations can help you find jobs that are located in your desired geographical area..

Disadvantages

- The app may not have all the skills you need in its database.

- The app may not be updated regularly, so you could end up with outdated information.

- The app may not be able to recommend skills based on your specific needs.
- The system may not be able to accurately recommend jobs that are a good fit for the user, leading to frustration and wasted time.

- Job recommendations may not be accurate or tailored to the user. The system may not be able to take into account the user's preferences or specific situation.

- The system may be biased. For example, if the system relies on input from previous employers, it may favor candidates who have already worked for the company.
- The system may be expensive to implement and maintain.

- The system may be abused. For example, users may try to game the system by entering inaccurate or misleading information.

11. CONCLUSION

In this project, we proposed a framework for skill/job recommendation application. This framework facilitates the understanding of the job recommendation process as well as it allows the use of a variety of text processing and recommendation methods according to the preferences of the job recommender system designer. Moreover, we also contribute making publicly available a new dataset containing job seekers profiles and job vacancies. Future directions of our work will focus on performing a more exhaustive evaluation considering a greater amount of methods and data as well as a comprehensive evaluation of the impact of each professional skill of a job seeker on the received job recommendation.

FUTURE SCOPE

The app could be expanded to include a social media component, where users could connect with each other and share tips and tricks. The app could also be expanded to include a gamification element, where users could earn points and badges for using the app frequently or for completing tasks.

There are many ways in which the skill recommender app could be improved. For example, the app could be made more personalized by taking into account the user's specific skills and interests. Additionally, the app could be made more interactive, perhaps by incorporating game-like elements or by allowing users to ask questions of the app's artificial intelligence (AI) system. Finally, the app could be made more comprehensive, perhaps by including a database of all known skills and by allowing users to search for skills by keyword.

12. APPENDIX

Source Code

```
from flask import Flask,render_template,request,session, redirect,url_for
import ibm_db
from dotenv import load_dotenv
import os

load_dotenv()

app = Flask(__name__)
app.secret_key= os.getenv("SECRET_KEY")

#Cloud Connection values
database_name = os.getenv("DATABASE")
host_name = os.getenv("HOSTNAME")
port = os.getenv("PORT")
uid = os.getenv("UID")
password = os.getenv("PASSWORD")

try:
    conn = ibm_db.connect(
        f"DATABASE={database_name};HOSTNAME={host_name};PORT={port
};SECURITY=SSL;SSLServiceCertificate=DigiCertGlobalRootCA.crt;UID={uid
};PWD={password}",'',''
    )
    print(conn)
    print("connection successful...")
```

```
except:
    print("Connection Failed")
    print(ibm_db.conn_error())



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

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

@app.route('/forgot')
def forgot():
    return render_template('forgotten-password.html')

@app.route('/signup', methods=['POST','GET'])
def signup():
    if request.method == 'POST':
        # conn = connection()
        try:
            sql = "INSERT INTO user_data VALUES('{}','{}','{}','{}')".format(request.form["name"],request.form["email"],request.form["phone"],request.form["password"])
            ibm_db.exec_immediate(conn,sql)
            #flash("successfully Registered !")
            return render_template('login.html')
        except:
          #flash("Account already exists! ")
          return render_template('signup.html')
    else:
          return render_template('signup.html')
        # name = request.form['name']
        #email = request.form['email']
        #phone = request.form['phone']
        #password = request.form['password']

        #sql ="INSERT INTO users VALUES (?,?,?,?)"
        #stmt = ibm_db.prepare(conn,sql)
        #ibm_db.bind_param(stmt, 1, name)
        #ibm_db.bind_param(stmt, 2, email)
        #ibm_db.bind_param(stmt, 3, phone)
        #ibm_db.bind_param(stmt, 4, password)
        #ibm_db.execute(stmt)
   # return render_template('signup.html')

@app.route('/login', methods=['POST','GET'])
def login():
    if request.method == 'POST':
      # conn =connection()
       email = request.form["email"]
       password = request.form["password"]
          sql = "SELECT COUNT(*) FROM user_data WHERE EMAIL=? AND PASSWORD=?"
       stmt = ibm_db.prepare(conn,sql)
```

```
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        res = ibm_db.fetch_assoc(stmt)
        if res['1'] == 1:
           session['loggedin'] = True
           session['email'] = email
           return render_template('job_post.html')
        else:
           #flash("email/ Password isincorrect! ")
           return render_template('login.html')
      else:
           return render_template('login.html')


        #email = request.form['email']
        #password = request.form['password']

        #sql = "SELECT * FROM users WHERE email=%s AND password=%s"
        #stmt = ibm_db.prepare(conn, sql)
        #ibm_db.bind_param(stmt,1,email)
        #ibm_db.bind_param(stmt,2,password)
       # user = ibm_db.execute(stmt).fetchone()

   #    return render_template('login.html' ,msg="success")

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

@app.route('/addrec',methods=['POST','GET'])
def addrec():
        arr = []
        sql = "SELECT * FROM job_list"
        stmt = ibm_db.exec_immediate(conn,sql)
        dictionary = ibm_db.fetch_both(stmt)
        while dictionary != False:
         inst={}
         inst['DNAME']=dictionary['JOBNAME']
         inst['DTITLE']=dictionary['JOBTITLE']
         inst['DROLE']=dictionary['JOBROLE']
         inst['DESCRIPTION']=dictionary['JOBDESCRIPTION']
         arr.append(inst)
         dictionary = ibm_db.fetch_both(stmt)

        return render_template('list.html',arr=arr)

@app.route('/list')
def list():
        arr = []
        sql = "SELECT * FROM job_list"
        stmt = ibm_db.exec_immediate(conn,sql)
        dictionary = ibm_db.fetch_both(stmt)
        while dictionary != False:
         inst={}
         inst['DNAME']=dictionary['JOBNAME']
         inst['DTITLE']=dictionary['JOBTITLE']
         inst['DROLE']=dictionary['JOBROLE']
         inst['DESCRIPTION']=dictionary['JOBDESCRIPTION']
         arr.append(inst)
         dictionary = ibm_db.fetch_both(stmt)
        print(arr)
        return render_template('list.html',arr=arr)


if __name__=='__main__':
      app.run(debug=True)
```

GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-17203-1659630277.git

https://drive.google.com/file/d/1CQ82f1c-Y2rtU3SXzQ7Z2PZw6h2FATJ-/view?usp=share_link